Extraktion von regelbasiertem Wissen aus Genexpressionsdaten und dessen Validierung

Dissertation
zur Erlangung des akademischen Grades
doctor rerum naturalium (Dr. rer. nat.)

vorgelegt dem Rat der Biologisch-Pharmazeutischen Fakultät
der Friedrich-Schiller-Universität Jena
von Vladimir Monossov
geboren am 23.07.1966 in Leningrad (St.Petersburg) in Russland

Gutachter

1.
2.
3.

Tag des Rigorosums:
Tag der öffentlichen Verteidigung:

Inhaltsverzeichnis

Abkürzungen

AID	Automatic Interaction Detection
ALL	Acute Leucoblastic Leukemia
AML	Acute Myeloid Leukemia
ANOVA	ANalysis Of VAriance
CART	Classification And Regression Trees
cDNA	Complementary DesoxyriboNucleic Acid
CHAID	CHi-square Automatic Interaction Detection
DNF	Disjunctive Normal Form
HTML	HyperText Markup Language
IG	Information Gain
JDBC/ODBC	Open Database Connectivity /Java Database Connectivity
JDOM	Java-based Document Object Model
k-NN	k-Nearest Neighbors
mRNA	messenger - RiboNucleic Acid
NB	Naive Bayes
NLP	Natural Language Processor
PV	Prediction Votes
QUEST	Quick Unbiased Efficient Statistical Tree
SCV	Comma Separated Values
SPLASH	Structural Pattern Localization Analysis by Sequential Histograms
SQL	Structured Query Language
SVM	Support Vector Machine
TCP/IP	Transmission Control Protocol / Internet Protocol
TF	Therapy Failure
TIMI	Text Influenced Molecular Indexing
XML	Extensible Markup Language

Liste der wichtigsten Symbole

Δn	Differenz der Stimmenzahlen für eine und die entgegengesetzte Klasse
A_i	Klasse
B_{free}	Bonferroni-Multiplikator
β_i^l	Regel
c	Anzahl von originalen Kategorien
C_k	Untermenge der Objekten
$D_i(\Delta n)$	Dispersion von Δn,
$IG(W, x)$	Information Gain
$\gamma(A_i)$	Wahrscheinlichkeit, dass das Objekt k zur Klasse A_i gehören wird
$\gamma_i(\beta_i^l)$	Maß der Kreuzung der Klasse A_i auf dem Gebiet der Richtigkeit der Regel β_i^l
k	Objekt
K	Klassenanzahl
Λ	Anzahl der richtig erkannten Objekte
μ	Mittelwerte
$M_i(\Delta n)$	Erwartungswert der Differenz der Stimmenzahlen
$M_j(n_i(k))$	Erwartungswert der Stimmenzahl für die Klasse A_i
N_t	Anzahl der Objekte im Zustand A_i.
N_{Li} oder N_{Hi}	Anzahl der Objekte mit dem Wert L oder H nur im Zustand A_i
N	Objektesanzahl
$n_i(k)$	Stimmenzahl für eine Klasse

p	Variablenanzahl
$P(C_i)$	Wahrscheinlichkeit (verhältnismassige Häufigkeit) der Klasse i in einem Satz
Q	Anzahl der disjunktiven Sätzen
σ	Standardabweichung
r	Anzahl von vereinten Kategorien
$r(\beta_i^l)$	Gewicht der Regel β_i^l
S	Informationsgehalt des Merkmales
S_c	Schwellenwertes des Informationsgehaltes
$\tau_i(k)$	Summe des Gewichtes
V	eine der möglichen Bedeutungen von x
W	Satz der Objekte
W_{xv}	Untermenge von W mit den Objekten, in welchen $x = v$
x	betrachtete Variable

1 Einleitung

1.1 Motivation

Die zunehmende Quantität und Qualität der Sequenzierungen von Genomen hat ein neues Gebiet der Genomforschung, das biologische Funktionen und DNA-Sequenzen verbindet, - die funktionelle Genomforschung, eröffnet. Innovative Technologien, wie cDNA- und Oligonukleotid-Arraytechnologien, sind entwickelt worden, um DNA-Sequenzdaten zu nutzen und Genexpressionsdaten für ganze Genome zu gewinnen [Lockhart1996, Griffiths1996 und Dudoit2000].

Da die Arraytechnologie die Messung der Expressionsintensitäten tausender Gene gleichzeitig ermöglicht, werden damit bedeutende Beiträge zum Fortschritt in den fundamentalen Fragen der Biologie und der Medizin erwartet [Dettling2002, Liu2003].

Eine auf Genexpressionsprofilen basierende Unterscheidung zwischen Phänotypen (Klassen), wie z.B. zwischen normalem und Tumorgewebe oder zwischen Krebsarten, hat das Potential, unser Verständnis von Genexpression verschiedener Krebszellen zu fördern [Nguyen2002a, Khan1998]. Solche Techniken führen zu einem verbesserten Verständnis der molekularen Tumorvariationen und damit zu ihrer feineren und informativeren sowie molekular begründeten Klassifikation [Dettling2002].

Es wird erwartet, dass bei der globalen Genexpressionanalyse von Krebszellen eine relativ kleine Anzahl von Genmarkern identifiziert wird, die die Genauigkeit der klinischen Tests verbessern werden [Alizadeh2000].

Eine zuverlässige und präzise Tumorklassifizierung ist für die erfolgreiche Krebsbehandlung wesentlich. Die Fähigkeit, zwischen bekannten oder unentdeckten Tumorklassen erfolgreich zu unterscheiden, ist somit ein wichtiger Aspekt dieser neuartigen Methode der Krebsklassifikation [Ross2000, Dudoit2000].

Obwohl auch viele nicht genetisch bedingten Faktoren, wie Umwelteinflüsse oder das Alter, das klinische Ergebnis beeinflussen, sind die molekulargenetischen Analyseergebnisse vielversprechend als Prognoseindikatoren basierend auf der Arrayanalyse. Die große Datenmenge, die bei der Arrayanalyse erhalten werden, stellt eine Herausforderung für die Datenanalyse dar [Thomas2001, Herzel2001].

1.2 Grundlagen der Arraytechnologien

Ein DNA - Array besteht aus Tausenden von DNA-Sequenzen. Sie sind auf einem Träger, z.B. einem Mikroskopie-Slide aus Glas fixiert. Die Häufigkeit der diesen DNA-Sequenzen (Genen) entsprechenden mRNA-Spezies kann durch vergleichende Messung der Hybridisierungsintensitäten verschiedener (z.B. zweier) Proben bewertet werden.

Die mRNA oder dazu komplementäre cDNA zweier Proben wird mit verschiedenen Fluoreszenzfarbstoffen markiert, gemischt und mit den DNA-Sequenzen (den Sonden) hybridisiert. Die Fluoreszenzmessungen erfolgen getrennt für jeden Arrayspot. Die Fluoreszenzintensität spiegelt die relative mRNA-Menge in diesen zwei Proben wider [Schena1995, Dudoit2000]. Wegen des Unterschieds bei der Probenbehandlung, der Markierung, der Färbung und der Messung kann die Fluoreszenz nicht unmittelbar, aber nach der entsprechenden Normalisierung verglichen sein [Huber2002].

Die so gemessenen Spotintensitäten und die Anzahl der verschiedenen mRNA - Kopien in den Proben sind ungefähr proportional. Die Proportionalitätskonstante zwischen gemessenen Werten und der

mRNA - Kopienzahl in der Zelle ist unbekannt. Diese Proportionalitätskonstante kann von Array zu Array verschieden sein. Um den Einfluss des unbekannten Proportionalitätsfaktors zu eliminieren, werden die Daten normalisiert, d.h. z.B. mit einem zu bestimmenden Koeffizienten multipliziert, und im Vergleich verschiedener Experimente analysiert (Berechnung so genannter Ratios).

Eine Vergleichanalyse der Genexpression in den gesunden und kranken Zellen kann für die Identifizierung von Krankheit-verursachende Gene als potentielle Ziele einer medikamentösen Therapie verwendet werden. Ein wichtiger Aspekt ist die Vorhersage der Tumorarten auf der Basis der Genexpressionsdaten [Dudoit2000, Zhang1997].

Molekulare Diagnostik ist eine wachsende Disziplin, die das Potential hat, sowohl präventive Medizin wie auch die Krankheitsbehandlung zu beeinflussen [Bijlan2003].

1.3 Mängel von bekannten Arrayanalysenmethoden

Nachfolgend werden die am häufigsten in der Arraydiagnostik verwendeten Methoden diskutiert:

- *Clusterbildungsmethoden*: Die Clusteranalyse benutzt man am häufigsten in der Arraydatenanalyse. Bei der Clusteranalyse werden die Gene oder die Objekte (z.B. Proben oder Patienten) aufgrund der Ähnlichkeit der Genexpressionsmuster gruppiert [Thomas2001, Yeung2001]. Mehrere Clusteranalysemethoden sind für die Analyse von Genexpressionsdaten verwendet worden [Eisen1998; Alon1999; Perou1999; Ben-Dor1999; Ben-Dor2000; Tibshirani1999].

  Die Clusteranalyse entdeckt Gene oder Proben, die ähnliche Expressionsprofile zeigen, und kann somit die Gene entdecken, die die Proben, z.B. gesundes Gewebe von krebsartigem Gewebe, unterscheiden können. Dabei ist die Identifikation von „ähnlichen" Gruppen notwendig. Die Clusteranalyse ist keine passende Methode für die Klassifikation, weil eine Clusteranalyse sich auf Gruppenähnlichkeiten konzentriert (unüberwachtes Lernen) und nicht die Merkmale vorgegebener Unterschiede sucht (überwachtes Lernen). Clusteranalysealgorithmen sind auch unfähig, vorexistierendes Wissen auszunutzen [DeRisi1997, Thomas2001].

- *K-Nearest Neighbors - k-NN* [Cover1967] und *Support vector machines - SVM* [Burges1998, Vapnik2000] wurden in [Furey2000, Brown2000, Lee2002, Sarkar2000] benutzt. *k-NN* ist einfach, intuitiv und hat eindrucksvoll niedrige Fehlerraten. In [Dudoit2000] wurden *SVM* für die Analyse von Leukämie-Daten verwendet, wobei jedoch keine besseren Ergebnisse erhalten wurden als mit *k-NN*. In [Brown2000] wurden *SVM* für die Gen-Klassifikation verwendet; sie zeigten bessere Ergebnisse als ein cross-validierter unaggregierter Entscheidungsbaum.

- *Lineare Diskriminanzanalyse* [Fisher1936, Fix1951, McLachlan1992], *Korrelationsmethoden* [Golub1999], *Native Bayes - NB* [Langley1992]: Obwohl diese Algorithmen unterschiedlich sind, kann man sie in einer Gruppe zusammenfassen, weil das Einteilungsprinzip sehr ähnlich ist. Bei diesen Methoden werden Wechselbeziehungen zwischen Genen ignoriert und die vorhergesagte Klasse hat einfach die maximale Wahrscheinlichkeit. Die Ergebnisse dieser Algorithmen sowie von *k-NN* und *SVM* sind schlecht interpretierbar.

  Diese Algorithmen und lineare Algorithmen sind sehr ähnlich. Die Besonderheiten der linearen diagnostischen Modelle, in der das diagnostische Ergebnis die Summe der Merkmale (Genexpression) ist, sind gut studiert. Sie basieren auf der Annahme der statistischen Unabhängigkeit der Merkmale, die in diesem Fall wenig begründet ist. Außerdem wird angenommen, dass die Merkmale keine Wechselbeziehung haben.

  Die Ergebnisse dieser Modelle werden als Entfernung der untersuchten Objekte von einer Hyperebene im Merkmalesraum oder die Objektesprojektionen an eine Linie im gegebenen Raum interpretiert. Deshalb können die linearen Modelle adäquat nur einfache geometrische Objekte des Merkmalesraumes, in dem die Objekte der verschiedenen diagnostischen Klassen dargestellt werden, beschreiben. Im Falle komplizierter Verteilungen können diese Modelle nur wenige Besonder-

heiten der Struktur der experimentalen Daten (ohne Vergrößerung der Merkmalesraumesdimension) widerspiegeln. Solche Besonderheiten sind fähig, wertvolle diagnostische Information zu tragen, deren Erscheinen in einer realen Aufgabe als Ausnahme gewertet werden kann. Im Allgemeinen entdeckt die Anwendung der linearen Modelle nur die „gröbsten" Besonderheiten der experimentalen Information [Lucenko1999].

Die Verwendung geometrischer Methoden bei einem medizinischen diagnostischen Prozess hat ein weiteres Problem. Wenn die Zahl der Lernobjekte und die Anzahl der Merkmale vergleichbar sind, kann die teilende Hyperebene praktisch immer (ohne Klassifikationsfehler auf dem Lernsatz) gefunden sein. In den medizinischen Aufgaben ist die Anzahl der verifizierten Daten sehr beschränkt und die Merkmalanzahl kann sehr groß sein. Man kann immer durch die Vergrößerung der Merkmalanzahl (bei gegebener Datenanzahl) die teilende Hyperebene finden, aber der Nutzen solcher Teilung ist gering, weil nicht generalisierbar.

Bei der Anwendung der geometrischen Methoden müssen die Bedeutungen aller Merkmale aller Objekte bekannt sein. In den medizinischen Aufgaben ist das nicht immer realisierbar [Carp1976].

- *Entscheidungsbaum*: Unter den Methoden zur Induktion von Entscheidungsbäumen sind *C4.5*[Quinlan1993] und *CART* [Breiman1984], die wahrscheinlich populärsten Algorithmen. In [Dudoit2000] ist gefunden worden, dass *CART*-basierende Modelle eine Güte haben, die zwischen linearen Diskriminanzanalyse-Methoden und *k-NN* liegen. Die Klassifikationsgenauigkeit von Entscheidungsbäumen kann durch die Aggregation verbessert werden. Verbundene Variablen sind im Allgemeinen genauere als ein Baum aus einzelnen Variablen, aber die Einfachheit und Interpretierbarkeit geht dabei verloren. Entscheidungsbäume werden weiter unten vertieft analysiert. Im Gegensatz zu den vorgenannten Methoden decken *Entscheidungsbäume* und *logische Regeln* Interaktionen zwischen den Merkmalen (Genen) auf. Bäume helfen die Beziehungen zwischen Vorhersagevariablen (Expression von Genen) und Antworten (Phänotyp, Krankheitszustand) zu interpretieren und diese Beziehungen (durch die schrittweise Auswahl der Variablen) finden.

- Logische Regeln und fuzzy Regeln sind in [Li2002, Guthke2002] benutzt. Sie werden jedoch selten angewandt, wahrscheinlich wegen der Extraktionskompliziertheit und der nicht hohen Erkennungsqualität der existierenden Algorithmen. Obwohl *C4.5* ähnliche Regeln erzeugen kann, sind diese Regeln nicht zuverlässig [Li2003].

Verschiedene *Data mining* Methoden werden gegenwärtig für die Genexpressionsanalyse genutzt, aber es existiert keine universell akzeptierte Methodik für die Analyse solcher Datensätzen [Bijlan2003]. Der Mangel an geeigneten statistischen Methoden der Arraydatenanalyse ist ein kritischste Problem, das der erfolgreichen Anwendung dieser viel-versprechenden Techniken in biologischen Forschungen entgegensteht [Szabo2002]. Die bisher benutzten Methoden sind nicht entwickelt worden, um die folgenden besonderen Anforderungen der Arrayanalysis zu erfüllen:

- In Arraydaten ist die Variablenanzahl p mehr als 100 Mal größer als die Objektesanzahl N (die öffentlich verfügbaren Datensätze enthalten Genexpressionsdaten für mehr als 5000 Gene und weniger als 100 Beobachtungen [Dudoit2000]). Übliche statistische Methodiken arbeiten gut, wenn $N > p$ [Nguyen2002a];

- Die bestehende Techniken ermöglichen nicht immer molekularbiologisch-medizinisch fachliche Einschätzung und Analyse der Ergebnisse; sie geben nicht immer eine Erklärung und damit kann die Bedeutung der Ergebnisse verborgen bleiben;

- Arraydaten haben große Messfehler, sie sind stark verrauscht und haben systematische Fehler. Die meisten der oben besprochenen Methoden lösen diese Problemen nicht hinreichend [Berrar2003, Robson2003].

Eine Modifikation der existierenden statistischen Methodiken oder eine Entwicklung neuer Methodiken ist für die Arraydatenanalyse notwendig [Nguyen2002a].

1.4 Mängel von bekannten Methoden der Mustererkennung und Wissensextraktion

Am Anfang der Methodikerarbeitung muss man die Forderungen analysieren. Die Mustererkennungssysteme sollen klare, für die Spezialisten verständliche Ergebnisse liefern. Es ist sehr schwer, die Richtigkeit der Beteiligung der Gene oder der Proteine in einem Prozess auf der Basis nur ihrer Bezeichnungen zu bewerten. Das Problem der Transformation des erhaltenen Wissens in eine klare Form ist extrem wichtig. Dieses Problem wurde zum Beispiel in [Oyama2002] untersucht: Die Autoren haben die Methode "Association Rules Discovery" vorgeschlagen, die die Wissensextraktion aus den Daten über die bekannten Protein-Protein-Wechselwirkungen, unterstützt. Die Methode hat es ermöglicht, die Regeln in einer natürlich - sprachlichen Form wie beispielsweise „*An SH3 domain binds to a proline-rich region*" zu entdecken. Diese Ergebnisse zeigen, dass neues Wissen entdeckt werden kann (wenigstens, bezüglich der Proteinwechselwirkungen).

Ein Arbeitsziel ist die Erarbeitung einer Methodik,

- die statistische Ergebnisse liefert,
- die unter Berücksichtigung des existierenden Wissens diese Ergebnisse analysiert und
- die das Wissen in einer klaren, für Menschen verständlichen Form vorstellt.

Für die Realisierung des Algorithmus ist die Entwicklung von zwei folgenden Methodiken notwendig:

- Methode zur Wissensextraktion;
- Wissensanalysetechnik.

1.4.1 Methode zur Wissensextraktion

1.4.1.1 Vorteile und Nachteile von logischen Algorithmen

Mustererkennungsmethoden konstruieren und nutzen formale Operationen für die Objektabbildungen, die die Äquivalenzbeziehungen zwischen diesen Objekten widerspiegeln. Die Äquivalenzbeziehungen drücken die Zugehörigkeit der bewerteten Objekte zu bestimmten Klassen aus, die als selbständige semantische Einheiten betrachtet werden können.

Das Erzeugen von funktionellen Zusammenhängen aus Lernbeispielen ist das „Lernen". Es gibt eine große Anzahl von Mustererkennungsmethoden und in der Literatur kann man die zahlreiche Klassifikationsmethoden finden[Ripley1996]. Ein wichtiger Vorteil der logischen Algorithmen ist die Darstellung der Ergebnisse in Form von logischen Regeln und diese Regeln können weiter durch Wissensanalysealgorithmen untersucht werden. Die logischen Regeln können damit beide Methodiken, d.h. Methode zur Wissensextraktion und Wissensanalysistechnik, miteinander verbinden

Die Erfahrung mit Lösungen von Mustererkennungsaufgaben lehrt, dass die entscheidende (unterscheidende) Information oft nicht in den einzelnen Merkmalen, sondern in ihren Kombinationen enthalten ist. In der molekulargenetischen Literatur kann man Hinweise darauf finden, dass Merkmalskombinationen für einige Erkrankungen charakteristisch sind und Gene miteinander wechselwirken. In [Lawrence1999] wird gezeigt, dass die Mitaktivierung von *HOXA9* und *MEIS1* in den *AML*-Fällen einen Weg vieler onkogener Mutationen darstellt.

Die logischen Mustererkennungsmethoden stützen sich auf den Apparat der Algebra und formalen Logik und können die Information, die sich in einzelnen Merkmalen und in Merkmalskombinationen befinden, bearbeiten. Beim Klassifizieren von Genexpressionsprofilen oder anderen Arten von biomedizinischen Daten sind einfache Regeln vorzuziehen [Li2003]. In diesen Methoden werden die Bedeutungen von Merkmalen als elementare Ereignisse betrachtet. Man kann die logischen Methoden als eine Art der logischen Suche im Lernsatz und als die Formulierung eines Systems logischer Regeln, von denen jede ein eigenes Gewicht hat, charakterisieren[Lucenko1999].

Ein wichtiger Unterschied der logischen Algorithmen zu anderen Klassen von Algorithmen ist die bedeutend schwächeren Anforderungen an die Ausgangsinformation (Verwertbarkeit von Daten geringerer Qualität). Die Daten können nicht nur in der Zahlenform (numerisch), sondern auch in natürlicher Sprache (deklarativ) beschrieben werden. Diese Algorithmen finden Diskriminanzfaktoren nicht nur in einzelnen Merkmalen, sondern auch der Merkmalskombinationen. Die Algorithmen können auch die Unterschiede einzelner Objekte der Datensatz berücksichtigen. Diese Klasse der Algorithmen eignet sich für statischen und dynamischen Aufgaben [Guthke1998].

Schon in der Mitte der 60er Jahre sind die ersten logischen Mustererkennungsalgorithmen entwickelt worden. Die logischen Mustererkennungsalgorithmen sind auf zwei große Gruppen aufzuteilen:

- die Algorithmen, die *DNF* (Disjunktiven Normalformen) konstruieren. Der bekannteste Algorithmus ist *Kora* (s.- folgendes Kapitel 1.4.1.2)
- die Algorithmen, die auf der Nutzung der Entscheidungsbäume basieren. Diese Algorithmen können auch Regeln generieren.

Nachfolgend werden diese Algorithmen getrennt diskutiert.

1.4.1.2 Kora

M.Bongard hat das Algorithmus *Kora* 1967 [Bongard1967] entwickelt. Seitdem wurde *Kora* erfolgreich in vielen Bereichen genutzt [Ochoa1998].

Es wird dabei angenommen, dass die Erkennung durch logische Funktionen ein Charakteristikum des menschlichen Denkens ist und dieses in gewissem Maße modelliert. Dies wird in den medizinischen Aufgaben besonders deutlich. Natürlich gibt es einen Unterschied zwischen dem menschlichen Denken, das die Aufgabe der Erkennung löst, und *Kora*. Der Algorithmus *Kora* ermöglicht die Vollsuche innerhalb der gegebenen Klasse der Beschreibungscharakteristiken, wobei die nützlichen logischen Funktionen extrahiert werden. Der Mensch kann die Regeln, die nach seiner Meinung für Diagnostik wichtig sind, „auswählen". Der Erfolg solcher Auswahl basiert (im wesentlichen) auf der Intuition und auf den gesammelten und unterbewussten Erfahrungen.

Kora ist frei von einer Reihe von Mängeln, die für andere Erkennungsmethoden typisch sind. *Kora* fordert, zum Beispiel, keine statistische Unabhängigkeit der Kennzeichen. *Kora* kann ohne Kenntnisse aller Koordinaten der die Objekte darstellenden Punkte im Merkmalsraum im Laufe des Lernens funktionieren und ermöglicht die Prüfung auch der unvollständig beschriebenen Objekte. Im Vergleich zu den geometrischen Methoden kann man eine geringere Fehleranzahl beim Lernen erwarten [Carp1976].

Algorithmus Kora analysiert alle Konjunktionen

$$T_1 \wedge T_2 \wedge T_l$$

$l <= L$,
T — die elementaren Ereignisse,
L — eine Zahl (im Algorithmus *Kora*3 $L = 3$).

Die Konjunktionen, die in einer Klasse öfter als einige Schwelle „Min. Richtig" richtig sind und seltener als die Schwelle „Max. Fehler" falsch sind, werden markiert. Um die neue Beobachtung k einzustufen, wird für diese Beobachtung die Anzahl $n_i(k)$ der Regeln, die für das Objekt k richtig sind (die für die Klasse i stimmen), berechnet. Wenn $n_i(k)$ ein Maximum über alle Klassen erreicht, wird das Objekt als Objekt der Klasse i klassifiziert.

Für die Induktion der Regeln wird beispielsweise der folgende Algorithmus verwendet:

```
FOR (INT i1 = 0; i1++; i1 < VariableNumber - L) {
        ...
        FOR (INT il= i1 + L-1; il++; il < M ) {

        /****************** Analyse der Konjunktion T1 ∧ …. ∧ T1 ********************/

        }
        ...
}
```

„VariableNumber“ ist die Anzahl der Variablen und *L* ist der Konjunktionsrang.

Ein offenbarer Mangel des Algorithmus ist sein Rechenaufwand, da er auf der vollen Suche aller Merkmalskombinationen basiert. Deshalb werden bei der Anwendung der logischen Methoden hohe Forderungen an die effektive Organisation des Rechenprozesses gestellt. Diese Methoden arbeiten bei verhältnismäßig kleinen Dimensionen des Merkmalsraumes und niedrigen Werten *L* (Konjunktionsrang) und auch dann nur auf Hochleistungsrechnern [Lucenko1999, Djuck2001]. Andererseits ist die direkte praktische Anwendung des Algorithmus mit dem niedrigen Konjunktionsrang 3 bei *Kora*3 auf medizinische Fragestellungen uneffektiv. Bei der Objektklassifikation symptomatisch ähnlicher Erkrankungen durch die *Kora*3 kann man nur wenig Konjunktionen finden, die nur für eine Klasse zutreffen. Die Konjunktionen klassifizieren nur einen kleinen Teil des Lernmaterials richtig ein. Nur solche „reinen“ (d.h. fehlerfrei klassifizierenden) Konjunktionen von Merkmalsbewertungen (Konditionen) sind für *Kora*3 wichtig [Carp1976, Ruiz-Shulcloper2000].

1.4.1.3 Entscheidungsbaumgenerierung

Das Ziel der top-down-Induktion der Entscheidungsbaume besteht darin, einen solchen Entscheidungsbaum zu finden, der am kleinsten ist und die Lerndaten richtig klassifiziert. Das Ergebnis der Algorithmusarbeit ist ein Entscheidungsbaum oder ein Satz der logischen Regeln. Die logische Regeln können interpretiert und mit dem existierenden Wissen verglichen werden. Das unterscheidet sie von anderen Algorithmen. Ein Entscheidungsbaum ist ein Baum, an dessen Verzweigungspunkte (innere Knoten) die Variableswerte getestet werden und dessen Blätter (äußere Knoten) die zu einer Klasse gehörenden Elemente enthalten, die also die Tests befriedigen. Ein Datensatz *W* wird durch den Algorithmus der Entscheidungsbaumkonstruktion auf *Q* disjunkten Sätzen $C_1, C_2, ..., C_Q$ verteilt:

$$W = \bigcup_{i=1}^{Q} C_i \quad \text{und } C_i \cap C_j = \text{null}. \tag{1.1}$$

Klassifikationsbäume unterscheiden sich von traditionellen statistischen Methoden der Klasseanalyse: Die Bäume verwenden eine Hierarchie von zur Objektsortierung geeigneten Variablen. Die Prüfung wird auf jedem Schritt durchgeführt. Bei traditionellen Methoden erfolgt diese Prüfung auf Klassenmitgliedschaft für jedes Objekt einmalig Die Klassifikationsbäume gehören zu den meisten verwendeten Methoden des überwachten Lernens und der Klassifikatorkonstruktion [Mitchell1997, Winston1992, Quinlan1986].

Nachfolgend werden einige der bekanntesten Methodiken diskutiert.

CART- ähnliche Methoden

CART (*Classification And Regression Trees*, [Breiman1984]) und *CART*-ähnliche Methoden untersuchen Baumverzweigungen, d.h. alle Verzweigungen von allen Vorhersagevariablen, um diejenige Verzweigung zu finden, die den größten Informationsgewinn (*InformationGain*) - *IG* produziert [Han2000, Quinlan1986].

$$IG(W, x) = I(W) - \sum_{v \in x} \frac{|W_{xv}|}{|W|} * I(W_{xv}), \quad (1.2)$$

x ist die betrachtete Variable,
W ist der Satz der Objekte,
v ist eine der möglichen Werte von x,
W_{xv} ist die Untermenge von W mit den Objekten, in welchen $x = v$
$I(W_{xv})$ und $I(W)$ ist die Entropie:

$$entropie(W) = -\sum_{i=1}^{K} p(C_i) \log p(C_i), \quad (1.3)$$

oder *Gini*-Index:

$$gini(W) = 1 - \sum_{i=1}^{K} p(C_i)^2, \quad (1.4)$$

Dabei ist $p(C_i) = |C_i| / |W|$ die Wahrscheinlichkeit (verhältnismäßige Häufigkeit) der Klasse i in diesem Satz und K ist die Klassenanzahl.

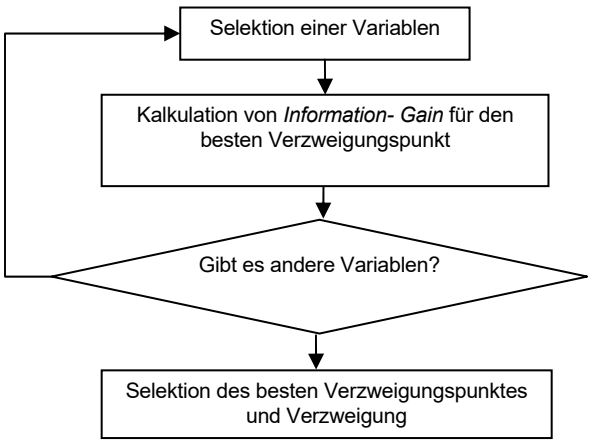

Abbildung 1.1. Die Verzweigungsselektion bei der *Information-Gain* -Nutzung

Das Ziel ist es, solche Teilmengen zu erhalten, die in Bezug auf die Klassenzugehörigkeit so homogen wie möglich sind. Dafür werden die Variablen verglichen und die Variable mit der besten Verbesserung hinsichtlich der o.g. Verzweigungskriterien für die Verzweigung ausgewählt. Der Prozess wiederholt sich rekursiv bis zur Erfüllung eines Stopkriteriums.

Die bekanntesten Algorithmen sind:

i.) *C4.5*
C4.5 [Quinlan1993] basiert sich auf *ID3* [Quinlan1979]. Der Algorithmus minimiert die durchschnittliche Entropie. *ID3* rechnet die durchschnittliche Entropie jeder Variable und wählt diejenige mit der niedrigsten Entropie.

ii.) *CART* ([Breiman1984])
Jede Verzweigung jeder Variablen wird eingeschätzt, um
- den besten Verzweigungspunkt oder
- die beste Gruppierung

zu finden.
CART basiert sich auf dem *Gini*-Index. Der *Gini*-Index erreicht den Wert Null, wenn in einem Knoten nur Objekte sind, die zu nur einer einzigen Klasse gehören. Und er erreicht seinen maximalen Wert, wenn die Verteilung der Objekte im Knoten über alle Klassen gleich ist. *CART* kann nur numerische Werte wirksam verarbeiten. *CART* - ähnliche Algorithmen können gute Einteilungen des Lernsatzes erzielen, aber diese Einteilungen sind häufig nicht generalisierbar, d.h. sie werden nicht unbedingt in Testdatensätzen richtig sein.

CHAID

CHAID (Chi-squared Automatic Interaction Detector) wurde 1980 von Kass entwickelt. *CHAID* macht eine erschöpfende Suche nach allen möglichen Teilmengen.

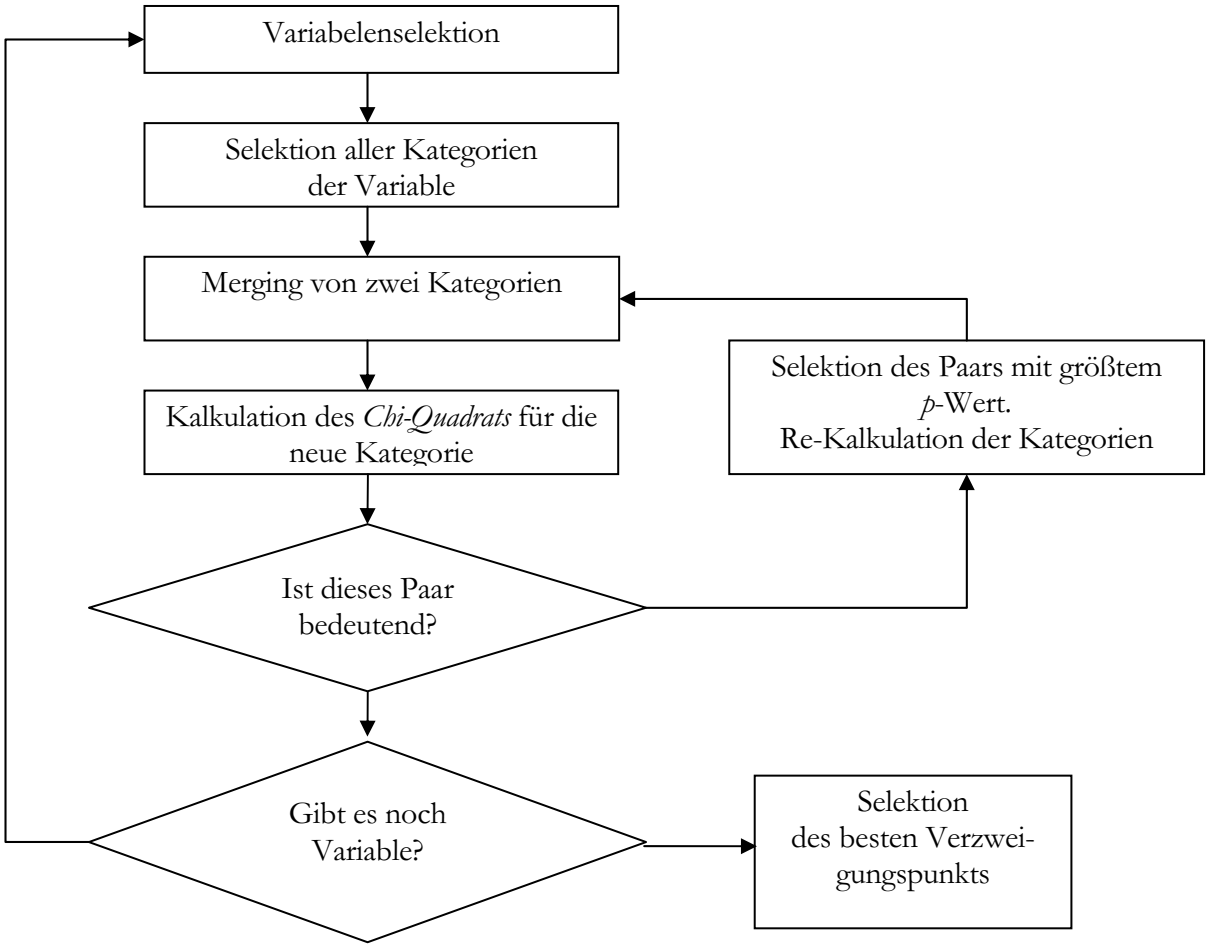

Abbildung 1.2. Die Verzweigung in CHAID

CHAID benutzt *Chi-Quadrat* als Verzweigungskriterium und für die vereinten Kategorien werden *Chi-Quadrat* - Werte durch einen Bonferroni-Multiplikator definiert. Der *Bonferroni-Multiplikator* wird wie folgt berechnet:

$$B_{free} = \sum_{i=0}^{r-1} (-1)^i \frac{(r-i)^c}{r!(r-i)!} \qquad (1.5)$$

c = Anzahl von originalen Kategorien und r = Anzahl von vereinten Kategorien.

Obwohl *CHAID* kompliziert ist, reduziert er die Fehler bei der Verzweigungsauswahl, die geschehen, wenn die *CART*-Such - Methode [Breiman1984] benutzt wird.

CHAID ist zuverlässig, jedoch langsam und beschränkt. *CHAID* spart keine Computerzeit und garantiert nicht, dass die gefundene Verzweigung das „Beste" ist.

QUEST

QUEST ("*Quick, Unbiased, Efficient Statistical Trees*") wurde 1997 von Loh und Shih[Loh1997] entwickelt. *QUEST* vermeidet Fehler bei der Variablenselektion durch Diskriminanz-basierende -Verzweigung.

QUEST führt die folgenden Schritte durch:

1. Für die Suche nach einer besten Variablen werden für jede Variable *p*-Niveaus berechnet und analysiert.
    a. Für kategorische (nominelle) Variable benutzt man *Pearson's Chi-Quadrat* Prüfungen.
    b. Für die kontinuierliche Variable wird ANOVA benutzt. Wenn das kleinste berechnete *p*-Niveau kleiner als *Bonferroni-adjusted-p-Niveau* ist, wird die Variable mit dem kleinsten *p*-Niveau gewählt, anderenfalls wird *Levene's F*-Test durchgeführt.

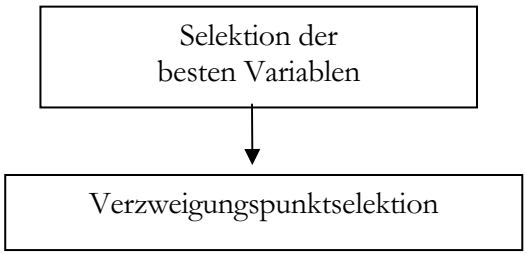

Abbildung 1.3. Die Verzweigung in QUEST

2. Die Verzweigung.
    a. Für die ordinale ausgewählten Variablen wird der Algorithmus von *Hartigan* und *Wong* (1979) angewandt, um zwei neuen Klassen zu schaffen.
    b. Für kategorische Variablen werden die Variablen in einen Satz der ordinalen Variablen umgewandelt und die Verfahren für ordinale Variable werden gestartet.

Der Prozess wiederholt sich rekursiv bis ein Stopkriterium erfüllt ist. Der Algorithmus ist kompliziert und spart keine Rechenzeit. Der Algorithmus garantiert nicht, die beste Verzweigung zu finden.

Wachsende Entscheidungsbäume

Wenn ein neues Lernbeispiel eingegeben wird, wird es durch den Entscheidungsbaum klassifiziert. Wenn es falsch klassifiziert wird, wird der Baum revidiert. Die Umstrukturierung des Baumes erfordert die Speicherung aller Beispiele oder das Beibehalten der Statistiken [Velde1989].

Es gibt Algorithmen, die praktisch benutzt werden, und solche, die nur theoretisch von Interesse sind. Bei komplexen mathematischen Modellen nehmen die Kompliziertheit der Programm-Realisierung und der Rechenaufwand zu. Damit sinkt die Chance, dass dieses System praktisch genutzt wird.

Häufig sind Algorithmen mit höherer Erkennungszuverlässigkeit auch komplexer. Das Modell muss einfach und praktisch wirksam sein[Lucenko1999]. Ein indirektes Kriterium der Algorithmusqualität ist die Programmrealisierung und Popularität dieser Programme. Diese Algorithmen (z. B. *D5R* oder *IDL*) werden nicht so oft als z.B. *C4.5* oder *CART* praktisch entwickelt und benutzt.

Aggregierte Entscheidungsbäume

In [Breiman1996] wurde es gefunden, dass die Genauigkeit der Klassifikation durch Variablenverbund gewinnt. Bei der Klassifikation können die Variablen durch Abstimmung (Voting) verbunden werden, d.h. wenn es eine Klasse gibt, die von einer (größeren) Anzahl von Variablen vorhergesagt wird.

In [Dudoit2000] sind Wiederholungen des Algorithmus benutzt worden und es wurde gezeigt, dass die Erhöhung der Anzahl von Wiederholungen von 50 bis 150 die Klassifikationsgenauigkeit nicht beeinflusste. Ein wesentlicher Mangel, des ansonsten positiv zu bewertenden Algorithmus ist der Verlust der Transparenz durch Aggregation von Variablen.

1.4.2 Text mining und andere Wissensverarbeitungstechniken

Biomedizinisches Wissen ist umfangreich in der Literatur und damit in Textform verfügbar. *Text mining* benutzt man, um neue Informationen aus existierenden Textdatenbanken herauszuziehen. Computermethoden analysieren Beziehungen zwischen Datenelementen und bringen neue Informationen über die bestehenden Daten hervor [Viator2001].

Es wurden Methoden der Informationsgewinnung entwickelt, um bestimmte Fachwörter, die im analysierten Text Objekten entsprechen, zu finden und zwischen diesen Beziehungen zu entdecken, also Verbindungen zwischen Fachwörtern zu extrahieren. Der Informationsgewinn besteht in der Entdeckung von komplizierten Strukturen in den Verbindungen von Fachwörtern. Diese Methoden interpretieren den Kontext, um die Bedeutung der biologischen Daten zu verstehen [Mack2002].

Es gibt eine große Zahl der Wissensextraktionsmethoden und Techniken für das *text mining* für die Arbeit mit existierenden Datenbanken.

Die Methoden des *literature mining* extrahieren und kombinieren Informationen aus wissenschaftlichen Veröffentlichungen. Viele Computerprogramme sind entwickelt worden, um verschiedene Informationen aus den in der Literaturdatenbank ‚*Medline*' gespeicherten *Abstracts* oder aus vollständigen Publikationstexten herauszuziehen [deBruijn2002]. Beispiele solcher Verfahren sind computer - unterstützte *Text mining* Programme für die Extraktion der Protein-Protein-Interaktionen [Albert2003] und der Genfunktionen [Chiang2003], *TIMI* [Singh2003], *ACROMED* [Pustejovsky2001].

Ein weiteres Beispiel *Metatool* [Schuster2003], das die Modellierung und Analyse metabolischer Netzwerke erlaubt.

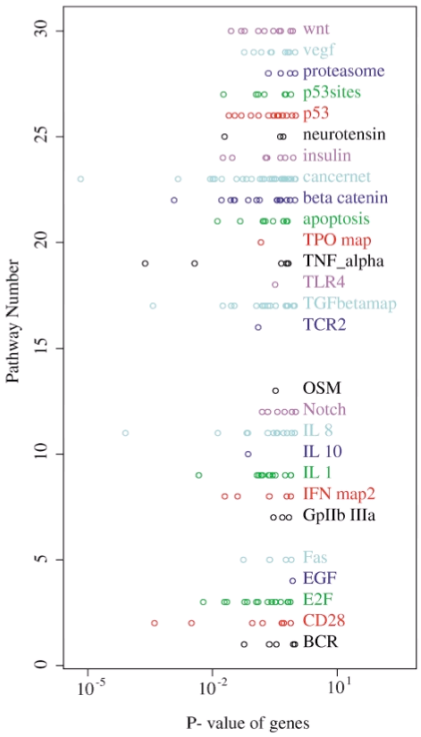

Abbildung 1.4. Ein Fragment vom GenePublisher Protokoll [Knudsen2002]

Nur wenige Techniken wurden bisher publiziert für die Analyse von experimentellen Ergebnissen aus den existierenden Datenbanken. Die Methodik der Durchführung der logischen Arraydatenanalyse ist in [Knudsen2003] vorgestellt. Die Daten werden auf dem Server zusammen mit der Spezifikation gesendet. Der Server macht die Normalisierung, die statistische Analyse und die Visualisierung der Daten. Die Ergebnisse werden mit Hilfe von Datenbanken für Promoter-Sequenzen, Signaltransduktionpathways und metabolischen Pathways analysiert. Die Analyseergebnisse werden in Form eines Protokolls zurückgesendet. Ein

Protokollbeispiel ist in Abb. 1.4 vorgestellt. Es ist kompliziert, eine praktische Lösung durch diese Technik zu finden. Vor allem bringt sie eine sehr große Zahl an Informationen, aber eine stringente Analyse wird nicht geleistet.

Die von [Knudsen2003] und [Oyama2002] eingeführten Techniken sind ungeeignet für die notwendige Wissensanalyse, weil sie für die Analyse von Ergebnissen von *unsupervised learning* Methoden entwickelt worden sein. Deshalb ist die Entwicklung neuen Algorithmen dringend erforderlich. Nach [Lucenko1999] können Mustererkennungsmethodiken in zwei große Gruppen aufgeteilt werden:

1. Methoden, die auf Operationen mit Merkmalen basieren:
   Bei der Konstruktion und der Anwendung der Algorithmen werden verschiedene Charakteristiken und Verbindungen von Merkmale verwendet. Dagegen werden die Objekte hier nicht als die informativen Einheiten betrachtet. Die Objekte treten als die Indikatoren zur Einschätzung der Wechselwirkung und des Verhaltens ihrer Attribute auf.

2. Methoden, die auf Operationen mit Objekten basieren:
   In diesen Methoden wird jedem Objekt ein selbständiger diagnostischer Wert gegeben. Im Unterschied zu den unter 1. genannten Methoden werden hier keine Information über einzelne Objekte ignoriert. Die Hauptoperationen bei diesen Methoden dienen der Bestimmung der Ähnlichkeit und der Unterschiede zwischen die Objekte. Die Objekte spielen die Rolle der diagnostischen Fälle.

1.5 Ziel der Arbeit

Für die Tumorforschung wurde von [Dudoit2000] gezeigt, dass es 3 statistische Probleme gibt.

1. Die Identifizierung von „Markergenen“, die verschiedene Klassen der Tumoren charakterisieren, d.h. die Auswahl von Variablen [Park2001]. Obwohl die Zahl der Gene (Variablen) in Tausenden gemessen wird, wird angenommen, dass nur wenige Gene die Klasse bestimmen [Dettling2002].

2. Die Krankheits-Klassifikation aufgrund bekannter Klassen (Diskriminanzanalyse / *supervised learning);*

3. Die Identifizierung von neuen, bisher nicht bekannten Tumorklassen aufgrund von Genexpressionsprofilen (Clusteranalyse / *unsupervised learning)*

   Zu diesen Problemen muss man noch ergänzen:

4. Analyse und Validation des regelbasierten Wissens durch existierende Wissensbasen.

Solche hier für die Tumorforschung vorgenommene Unterteilung der Methoden ist gültig auch für andere biomedizinische Forschungsgebiete. Ausgehend von den oben genannten vier Problemen kann man folgende Richtungen für die Methodikerarbeitung definieren:

1. die Methodik zur Diskretisierung und Selektion der Merkmale;

2. die Methodik zur Extraktion der regelbasierten Wissen;

3. Methodik zur Validation des regelbasierten Wissens.

Nach der Bewertung vorhandener Algorithmen kann man schlussfolgern, dass Logische Algorithmen zur Lösung aller drei Probleme geeignet sind. Sie

1. sind fähig, mit nahezu beliebigen Datenformaten zu arbeiten (Universalität);

2. sind gegen zufällige Messfehler (Rauschen) weniger empfindlich als andere Algorithmen (Robustheit);

3. erzeugen klar verständliche Ergebnisse (Transparenz).

Existierende logische Algorithmen haben eine Reihe von Mängeln. Der Algorithmus *Kora* macht keine volle Analyse aller relevanten Kombinationen von Merkmalen. Die Einführung von Beschränkungen, z.B. des Konjunktionsranges, kann die Ergebnisse wesentlich verschlechtern. Die populärste Algorithmen der Baumkonstruktion verwenden die relativ einfachen Algorithmen, die nicht immer genaue Ergebnisse geben. Versuche, die Arbeit der Algorithmen zu verbessern, führen zu komplizierten und praktisch nicht realisierbaren Methoden. Diese Algorithmen bauen prinzipiell nur einen einzigen Baum, der nicht obligatorisch der bestmögliche ist. Besonders im Falle kleiner Objektzahl führen die resultierenden Bäume zu nicht generalisierbaren Klassifikatoren. Im Fall einer großen Merkmalzahl kann die Konstruktion des optimalen Baumes praktisch unmöglich sein. Die Auswahl eines einzigen Baumes ist eine Schwäche der Entscheidungsbaum-Algorithmen, die vermieden werden muss. Der Versuch, das Problem durch Erzeugung aggregierter Bäume zu lösen, führt zum Verlust der Durchsichtigkeit und der Verständlichkeit der Ergebnisse.

Es ist die Erarbeitung einer neuen Regelgenerierungsmethode erforderlich. Der Schlüssel der Genauigkeitsverbesserung ist die mögliche Instabilität der Vorhersagemethode, d.h. dass die kleine Veränderungen im Lerndatensatz zu den großen Veränderungen im Klassifikator (Prediktor) führen. Entscheidungsbäume sind potenziell instabile Klassifikatoren, während andere Algorithmen, wie z.B. *k-NN* stabiler sind. Die instabilen Prozeduren profitieren durch die Aggregation [Clare2002]. Auch in [Clare2002] ist solches Herangehen zur Lösung des Problems untersucht. Die Aggregation führt jedoch zu komplizierten Rechenprozessen mit erhöhtem Rechenaufwand. Die nachfolgend vorgeschlagene Nutzung der Methodik der direkten Regelsatzerzeugung kann die Arbeit wesentlich vereinfachen.

Für die Analyse der erzeugten Regelsätze ist eine Methode zur Merkmalsanalyse, hier insbesondere der Proteininteraktion, erforderlich.

2 Die Entwicklung der neuen regelbasierten Wissensextraktionsmethode

2.1 Diskretisierung und Selektion von Merkmalen

Die Merkmalsselektion ist für jedes diagnostisches System wichtig [Bellamy1997]. Für die Nutzung von logischen Regeln ist darüber hinaus eine Datendiskretisierung notwendig. Weil traditionelle Methodiken der Datendiskretisierung (zum Beispiel, die Teilung des Werteumfangs der Merkmale in Intervalle) eine geringe Effektivität zeigen, werden neue Methoden der Datendiskretisierung und der Merkmalsselektion (zur Verkleinerung der Such-Raumesdimension) entwickelt. Für die Datendiskretisierung werden Mittelwerte und Standardabweichungen verwendet. Vor der Bestimmung dieser Größen werden zur Elimination möglicher Ausreißer die maximalen und minimalen Werte eliminiert. Aufgrund der verbleibenden *p*-2 Werte werden Mittelwerte μ und Standardabweichungen σ nach den Formeln 2.1 bzw. 2.2 berechnet:

$$\mu = \frac{\sum_{i=0}^{p} x_i - x_{\min} - x_{\max}}{p-2} \qquad (2.1.)$$

$$\sigma^2 = \frac{1}{p-3}\left(\sum_{i=0}^{n}(x_i-\mu)^2 - (x_{\min}-\mu)^2 - (x_{\max}-\mu)^2\right) \qquad (2.2.)$$

Die Diskretisierung wird entsprechend Abb. 2.1 durchgeführt. Dabei werden die Werte x markiert mit qualitativen (nominalen) Werten

„niedrig“, wenn $x < \mu - \sigma$

„normal“, wenn $\mu + \sigma \geq x > \mu - \sigma$

„hoch“, wenn $x > \mu + \sigma$.

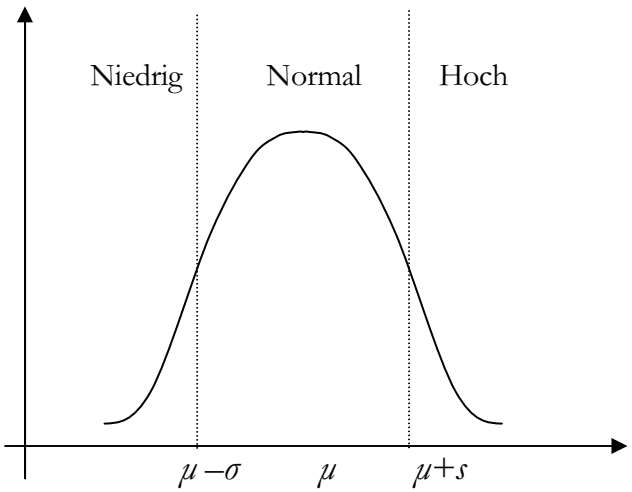

Abbildung 2.1. Datendiskretisierung

Im Fall der Normalverteilung sind im Intervall von μ - σ bis zu μ + σ 69%, und ober- sowie unterhalb diese Intervalle je 15,5 % aller Objekte (Lernbeispiele)

Die Diskretisierung dient auch der Auswahl der wichtigsten Merkmale.
Unter „wichtigen", d.h. informativen Merkmalen werden hier solche verstanden, die sich je nach dem Zustand (Phänotyp, Krankheitszustand) des Patienten hinsichtlich der o.g. qualitativen Merkmalswerte unterscheiden. Bei wichtigen Merkmalen sind die Werte „niedrig" oder „hoch" häufig auftretend in einem Zustand A_i (zum Beispiel, A_1 = „Geschwulst") und „normal" oder „hoch" bzw. „niedrig" im anderen Zustand (z.B. A_2 = „Tumorsabwesenheit").

Wenn sich die Werte eines Merkmals immer in einem Gebiet („niedrige", „normale" oder „hohe" Werte) befinden, ist dieses Merkmal nicht wichtig. Für die Einschätzung des Informationsgehaltes werden für jeden der Zustände die realisierten Merkmalswerte bestimmt. Dann wird die Menge der bei verschiedenen Zuständen nicht wechselnden Merkmalswerte bestimmt. Der prozentuale Anteil der Objekte, bei denen der Merkmalswert für verschiedene Zustände verschiedene qualitative Werte annimmt, wird als Informationsgehalt *S* definiert. Zum Beispiel, trifft der Merkmalswert „niedrig" und „normal" nur bei Kranken, und „normal" und „hoch" nur bei Gesunden zu. Es ist dann offensichtlich, dass die Werte „niedrig" für die Kranken und „hoch" für die Gesunden typisch sind und dieses Merkmal für die weitere Arbeit wichtig ist.

Informationsgehalt des Merkmales wird nach den Formeln 2.3 berechnet

$$S = \max (N_{L1} + N_{H2}, N_{L2} + N_{H1}) / N_t \qquad (2.3)$$

N_t ist die Anzahl der Objekte im Zustand A_i.
N_{Li} und N_{Hi} ist die Anzahl der Objekte mit dem Wert L oder H nur im Zustand A_i.

Bei der Merkmalauswahl wird ein Schwellenwert des Informationsgehaltes eingeführt. Es werden Merkmale gesucht mit Informationsgehalten größer 10 %, 20 %, … 90 %. Je höher der Informationsgehalt ist, desto wichtiger (interessanter) die selektierten Merkmale sein sollen, desto größer ist die Schwelle S_c zu wählen. Die Menge der wichtigen Merkmale verringert sich mit der Vergrößerung des Schwellenwertes des Informationsgehaltes S_c.

Die hier eingeführte Diskretisierungsmethode hat folgenden Vorteil: Wegen der technischen Schwierigkeiten ist die Relation (z.B. Proportionalitätskonstante) zwischen den gemessenen Größen (z.B. Fluoreszenzintensität eins Spots auf dem Mikroarray) und der Zahl der mRNA - Kopien in der Zelle unbekannt, und sie kann sich bei den verschiedenen Mikroarrays ändern. Deshalb werden die Daten üblicherweise normalisiert [Zien2001, Dougherty1995, Fayyad1993]. Solche Normalisierungsmethodiken sind relativ willkürlich motiviert bzw. kompliziert und garantieren nicht immer positive Ergebnisse. Bei Anwendung der hier eingeführten Diskretisierungsmethode werden keine absoluten quantitativen Merkmalswerte verwendet sondern ihre relativen, qualitativen Werte.

2.2 Regelextraktionsmethode

2.2.1 Bestandteile

Wie oben gezeigt worden ist, ist eine neue Methode zur Wissensextraktion aus Arraydaten erforderlich. Das vorliegende Kapitel ist der Entwicklung dieser neuen Technik gewidmet. Unter bekannten Techniken der Entscheidungsregelgenerierung können Algorithmen der Entscheidungsbaumkonstruktion und der *DNF*-Erzeugung (*DNF = Disjunktiven Normalform*; am populärsten ist der Algorithmus *Kora*) ausgewählt werden. Regeln sind ein „Nebenprodukt" der Entscheidungsbaumkonstruktion. Nachfolgend werden detailliert diese zwei Techniken betrachtet.

Der allgemeine Algorithmus zur Konstruktion von Entscheidungsbäumen wird in Abb. 2.2-2.3 gezeigt. Der Algorithmus spaltet die Daten hinunter bis ein bestimmtes Kriterium erfüllt ist, macht dann die Bewertung, die Optimierung und das Stutzen („*pruning*“) des Baums. Der Prozess stellt eine Wiederholung der Verzweigung dar und in einigen Fällen erfolgt auch eine Vereinigung (*„merging“*) vor der Aktivierung des Stopkriteriums. Nachfolgend werden die Möglichkeiten der gewöhnlichen Baumkonstruktionsmethoden bezüglich ihre Verwendung in der neuen Regelextraktionsmethode diskutiert.

Die folgenden Parameter werden in den Algorithmen benutzt:

1. **„Minimum_n“.** Die Verzweigung wird fortgesetzt, bis alle Blätter (finalen Knoten) des Baumes jeweils einer Klasse zugeordnet sind oder großer als ein spezifiziertes „Minimums n“ von Fällen enthalten. Ein „unreiner“ Knoten (d.h. ein Knoten, dem Objekte aus mehr als einer Klasse zugeordnet sind) muss „Minimum_n“ Objekte haben um gespalten zu werden
2. **Die Objektfraktion** Die Verzweigung wird beendet, wenn alle Blätter, die Objekte aus mehr als einer Klasse enthalten, nicht mehr Fälle als die ‚Objektfraktion‘ je Klasse haben. Die Objektenfraktion kann wie das zuvor genannte „Minimum n“ bestimmt sein.

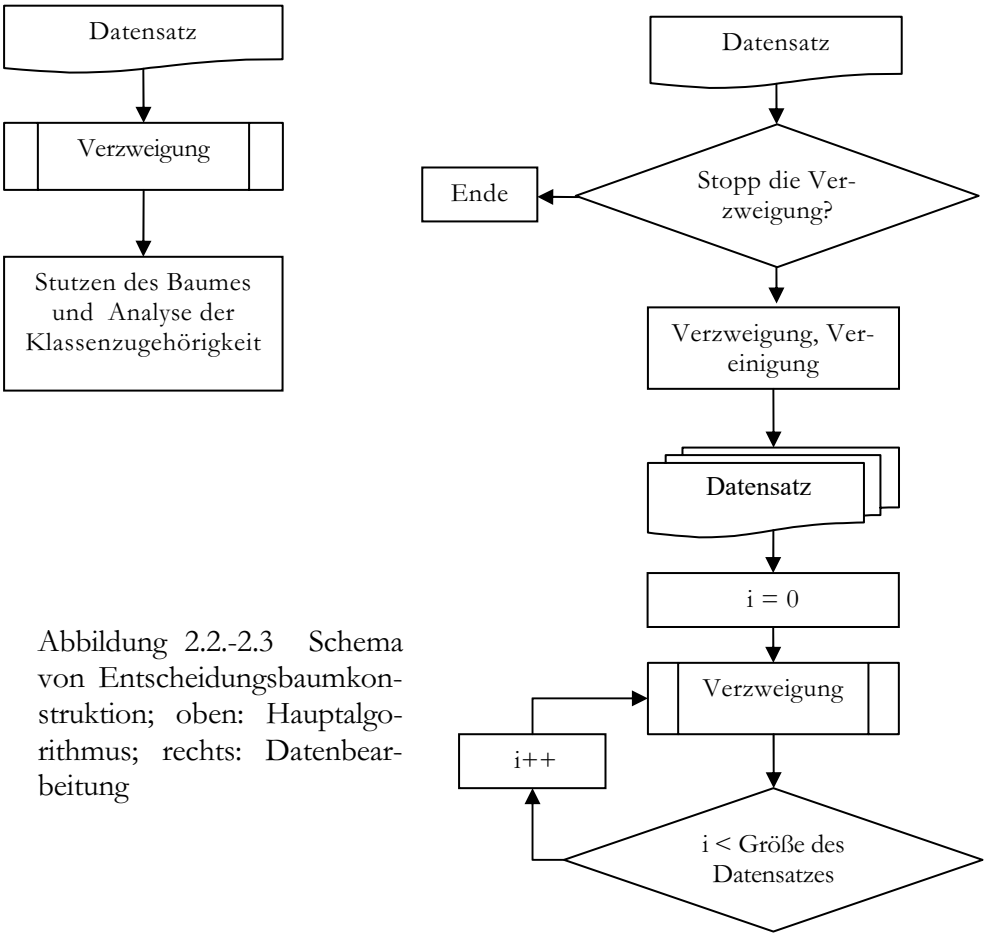

Abbildung 2.2.-2.3 Schema von Entscheidungsbaumkonstruktion; oben: Hauptalgorithmus; rechts: Datenbearbeitung

Diese Parameter sind die wichtigsten bei der Mehrheit der Algorithmen zur Entscheidungsbaumkonstruktion. Bei Abwesenheit der vorgenannten zwei Parameter arbeitet der Algorithmus ohne Unterbrechung bis zur vollen Teilung aller Objekte und eine wichtige Möglichkeit zur Optimierung des Baumes geht verloren. Bei kleiner Anzahl der Objekte führt ihre Analyse zu unerwünschter Baumgröße (mit Vergrößerung des Konjunktionsranges). Wenn die beiden o.g. Parameter zu groß sind, führt dies zu erhöhter Anzahl der Fehler des ersten Typs (falsch Negative). Wenn die Parameter zu niedrig sind, nimmt die Anzahl der Fehler des zweiten Typs (falsch Positive) zu. Das Auffinden des optimalen Wertes der vorliegenden Parameter ist eine der kompliziertesten Aufgaben der Baumkonstruktion. Diese Parameter werden bei der Erarbeitung der neuen Methodik der Konstruktion der logischen Regeln genutzt. Folgende zwei

(3. und 4.) Parameter, die in Algorithmen für die Entscheidungsbaumkonstruktion genutzt werden, werden dagegen im neuen Algorithmus nicht angewandt:

3. **Die Vereinigung (das *merging*) von unwichtigen und wichtigen Variablen** wird nur in einigen verfeinerten Algorithmen benutzt. *Merging* kann auf das Ende der Arbeit verschoben werden, also mit der Durchführung des Baum-Stutzens (*pruning*) verbunden werden.

4. **Verzweigung (*Split*)** ist die Selektion der Verzweigungspunkt bei Anwendung eines ‚Splitkriteriums'. Es gibt viele Verzweigungstechniken, die in zwei große Gruppen zu unterteilt werden können:

   - Multiverzweigung (macht k - 1 Verzweigungen, k ist die Anzahl der Variablen). Einige von diesen Algorithmen sind originell. *CHAID* wurde bereits oben dargestellt. Einige andere Algorithmen sind:
     - *FACT* (*Loh* & *Vanichestakul* in 1988)
     - *THAID* (*Morgan* & *Messenger* in 1973)
     - *AID* (*Morgan* & *Sonquist* in 1963)
   - Binäre Verzweigung (macht nur eine Verzweigung). Diese Art wird am meisten benutzt.

   Multiverzweigung hat keinen Vorteil, weil eine beliebige Multiverzweigung als Folge von binären Verzweigungen dargestellt werden kann. In den Multiverzweigungen können die Variabeln nur einmal zur Verzweigung dienen, so dass die Entscheidungsbäume zu kurz und uninteressant sein können [Ripley1996].

   Einige Algorithmen sind oben beschrieben worden. Die Verzweigungstechnik bestimmt Ergebnisse einer Baumkonstruktion. Das Wählen von verschiedenen Techniken führt zu unterschiedlichen Bäume. Die Wahl eines ‚Verzweigungskriteriums' (Splitkriteriums) ist für alle Baumkonstruktionsalgorithmen obligatorisch und ist ihre Schwäche. Bei jeder Phase einer Baumkonstruktion ist es notwendig, die Entscheidung einmalig zu treffen und diese Entscheidung sollte natürlich korrekt sein. Wenn alle Objekte gut trennbar sind, gibt es dabei keine Probleme. Aber es können Objekte aus verschiedenen Klassen fast identisch sein. Bei irgendeiner Phase der Konstruktion des Baumes können dann Bewertungen erfolgen die für mehrere Variable praktisch gleich sind. Die Notwendigkeit einer Entscheidungsauswahl kann in diesem Fall zu einem Fehler oder eine zufällige Auswahl einer Variablen führen. Besonders schwerwiegend ist dies Problem am Anfang einer Baumkonstruktion. Es ist offensichtlich, dass das gegebene Problem direkt von der Variablen- und Objektesmenge abhängt.

   Im unten untersuchten fiktiven „Gesichts" - Beispiel gibt es 16 Objekten und 16 Variablen. Sogar in diesem Fall gibt es die Situation, dass eine Wahl einer Splitvariablen unter mehreren gleich bewerteten erfolgen muss. In realen Fragestellungen, z.B. Bei der Analyse von Arraydaten, gibt es derartige Situationen mit dutzenden Objekten und Tausenden Variablen. Eine Baumkonstruktion ohne die Dimensionsraumverkleinerung der Variablen ist dann ein Zufall.

   In der hier vorgeschlagen neuen Technik der Regelgenerierung wurde auf die Wahl eines Verzweigungskriteriums verzichtet.

5. **‚Stopkriterien'** sind für die Arbeit eines jeden Algorithmus notwendig. Die Algorithmen der Baumkonstruktion verwenden folgende Stopkriterien:

   - Es wurden alle wichtigen (Auswahl siehe oben) Variablen verwendet
   - Die Anzahl der Objekte je Knoten ist kleiner oder gleich dem Parameter „Minimum_n"
   - In einem Knoten sind mehr Objekte verschiedener Klassen als der Parameter „Objektfraktion".

   Im hier entwickelten neuen Algorithmus werden ähnliche Stopkriterien verwendet.

6. ***Pruning* (Stutzen)** ist die Entfernung der für die Vorhersage unwichtigen Zweige. *Pruning* ist ein wichtiges Element im Prozess der Baumkonstruktion. Dies wird näher diskutiert:

Der Klassifikationsbaum „richtiger“ Größe ist ein solcher Baum, der hinreichend komplex ist für eine gute Klassifikation und dennoch so einfach wie möglich. Die kürzesten Bäume sind nicht immer die korrektesten [Mitchell1997, Winston1992]. Das Problem der „richtigen“ Baumgröße wird gewöhnlich durch das folgende Herangehen entschieden:

- Es wird von einem Fachexperten (Anwender) gelöst, wobei das bereits vorliegende Wissen , z.B. diagnostische Informationen von vorherigen Analysen oder sogar „Vorahnungen“, benutzt werden. Damit definiert der Fachexperte die Parameter der Stopkriterien für das Stutzen.
- Automatische, objektivierende Prozeduren, die in [Breiman1984] entwickelt worden sind, sind nicht fehlerfrei. Aber sie vermeiden das subjektive Moment im Prozess der Auswahl des „richtig großen“ Baumes. Zu diesen Prozeduren gehören insbesondere die *Cross-Validierungen*.

In [StatSoft1998] werden einige Varianten der *Cross–Validierung* vorgestellt:

- Testssatz *Cross-Validierung*.
- *V-fold Cross-Validierung.*
- *Global Cross-Validierung*

Diese drei Techniken sind unterschiedlich, aber die Probleme von Cross - Validation sind gleich: die Notwendigkeit der Auswahl einer einzigen Schlussfolgerung aus möglicherweise einigen gleich bewerteten. Es gibt keine Garantie, dass der gewählte Baum korrekt ist. Besonders wird es kompliziert, wenn die Unterschiede zwischen der Erkennungsqualität der konstruierten Bäume nur gering sind.

Die *Cross-Validierungen*, besonders im Falle der am effektivsten *V-fold* und *Global Cross-Validierung*, sind sehr rechenzeitaufwändig und können damit unpraktikabel sein [Dujk2002].

Ein Algorithmus, der detaillierter betrachtet werden soll, weil er Ausgangspunkt für den neu entwickelten Algorithmus ist, ist der Algorithmus *Kora*. Das Arbeitsschema des Algorithmus war oben untersucht (Kapitel 1.4.1.2). Der Algorithmus verwendet folgende Parameter:

1. ‚Min. Richtig‘ ist die minimale Zahl der Objekte, die eine logische Regel richtig klassifiziert. Der Parameter ist obligatorisch.
2. ‚Max. Fehler‘ ist die maximale Zahl der Objekte, die eine beliebige logischen Regeln falsch einstuft. Der Parameter ist auch obligatorisch.
3. ‚Konjunktionsrang‘ der erzeugten logischen Regeln ist die maximal erlaubte Anzahl von UND - Verknüpfungen in einer Regel. Für den Algorithmus *Kora* ist dieser Parameter obligatorisch. Er entspricht der maximalen Baumgröße der Algorithmen der Baumkonstruktion.
4. ‚Max. Korrelation‘ für zwei Konjunktionen dient der Auswahl von redundanten und somit, überflüssigen‘ Konjunktionen. Wenn der Korrelationskoeffizient zwischen zwei gewählten Konjunktionen größer als der Parameterwert ‚Max. Korrelation‘ ist, wird die „beste“ Konjunktion gelassen.
5. ‚Max. Anzahl‘ eines Merkmals in den Konjunktionen beschränkt die Konjunktionen mit ein und demselben Merkmal (aber unterschiedlichen Werten).
6. Die maximale Zahl der Konjunktionen über alle extrahierten logischen Regeln beschränkt die Anzahl der UND - Verknüpfungen summiert über alle Regeln.

Ein Hauptmangel des Algorithmus *Kora* ist die Notwendigkeit der vollen Suche und der Analyse aller Kombinationen der Merkmale, die die Bedingungen 1 und 2 befriedigen. Dabei kann eine sehr große Zahl von logischen Regeln erhalten werden. Deshalb kann die Regelextraktion ohne Beschränkung des Suchgebietes durch Einführung der zusätzlichen Parameter 3 bis 6 effektiv nicht geleistet werden. Der Parameter 3 ist in *Kora* obligatorisch. Aufgrund des großen Umfanges der Suche und damit des Rechenaufwandes wird im Algorithmus *Kora* der Konjunktionsrang gewöhnlich auf 3 begrenzt. Aber aus der Erfahrung bei der

Anwendung auf medizinische Probleme ist aber bekannt, dass der Konjunktionsrang 9 und mehr erforderlich sein kann [Carp1976]. Krone [Krone1999] schlägt einen Algorithmus zur Reduktion des Suchraumes vor, der auf der Häufigkeit von Objekten innerhalb des Suchraumes basiert.

Im hier neu entwickelten Regelextraktionsalgorithmus wird keiner der letzten 3 Parametern (4 - 6) verwendet.

2.2.2 Regelextraktion

Nach der Betrachtung der beider Gruppen der Regelextraktion, d.h. der Entscheidungsbäume und *Kora*, können folgende Schlussfolgerungen über die Besonderheiten dieser Algorithmen als Grundlage für die Entwicklung der neuen Methode getroffen werden:

1. Die im Algorithmus *Kora* implementierte Herangehensweise zur Generierung von logischen Regeln kann man im neuen Algorithmus verwenden. Dabei soll im neuen Algorithmus aber auf die volle Suche aller im Raum aller möglichen Kombinationen von Merkmalen verzichtet werden [Lucenko1999]. Die von *Kora* verwendeten zusätzlichen Beschränkungen (z.B. des Konjunktionsranges) sollen ebenfalls nicht angewandt werden.
2. Für die Entscheidungsbaumkonstruktion gibt es eine Zahl möglicher Splitkriterien. Da jeder der Verzweigungspunkt als eine Konjunktion von Merkmalen betrachtet sein kann, können die Methoden der Entscheidungsbaumkonstruktion auch für Erarbeitung des neuen Algorithmus zur Regelextraktion verwendet werden. Dabei soll aber der Mangel des Erzeugens nur eines einzigen Baumes vermieden werden.

Der neue Algorithmus zur Erzeugung die logischen Regeln verwendet folgende Spezifikationen bzw. Parameter (die Bedeutung der Parameter ist in Abbildung 2.4 dargestellt):

1. **Die Klasse**, für die die Suche durchgeführt wird. Ohne diesen Parameter kann man die logischen Regeln für beide oder alle Klassen suchen, wobei der Rechenzeitaufwand unnötig groß ist. Bei einem hinsichtlich Merkmal- und Objektanzahl großen Lernsatz kann der Rechenzeitaufwand kritisch groß sein. Dieser Aufwand kann bei Beschränkung auf eine vorgegebene Ziel-Klasse reduziert werden. Die ‚Klasse' als Parameter wird weder bei der Baumkonstruktion noch in *Kora* verwendet.
2. Die minimale Anzahl der Objekte, genannt ‚**Min. Richtig**', die von einer Regel korrekt klassifiziert werden müssen. Dieser Parameter entspricht dem Parameter ‚Min. Richtig' des Algorithmus *Kora*.
3. Die maximale Anzahl der Objekte, genannt ‚**Max. Fehler**', die vom einer Regel falsch klassifiziert. Der Parameter entspricht dem Parameter ‚Max. Fehler' des Algorithmus *Kora*.

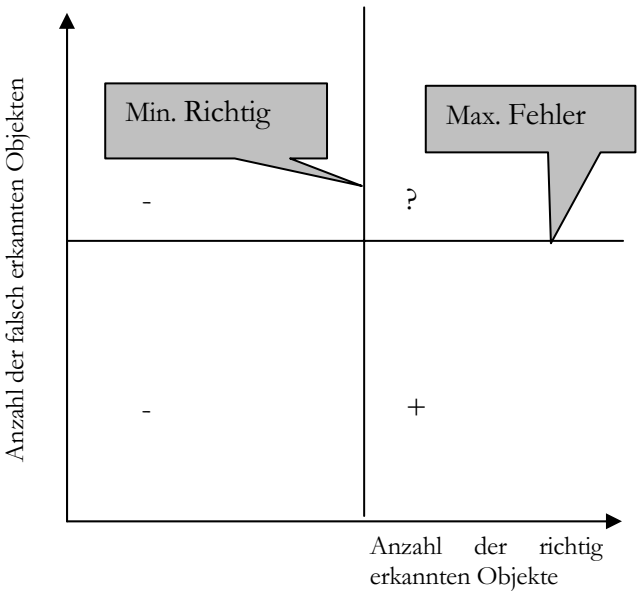

Abbildung 2.4. Grafische Erklärung der benutzten Parameter.

Auf der Abszissenachse sind die Objektanzahlen, die durch die logischen Regeln richtig erkannt sind. Auf der Ordinatenachse sind die Objektanzahlen, die durch die logischen Regeln falsch erkannt sind. Die Schwellenwerten ‚Min. Richtig ' und ‚Max. Fehler ' (die Parameter 2 und 3) sind als senkrechte bzw. waagerechte Linien dargestellt. Diese zwei Linien teilen die Fläche auf in vier Gebiete. Jede Regel kann einem der vier Gebieten zugeordnet werden.

Der Algorithmus hat das Ziel, solche logische Regeln zu finden, die oft (mehr als ‚Min. Richtig') zutreffen für die gewünschte Klasse und selten (weniger als ‚Max. Fehler ') zutreffen für die andere(n) Klasse(n). Wenn eine Regel in diesem mit ‚+' markierten Gebiet liegt, befriedigt diese Regel die Bedingungen der Suche. Die Regel wird protokolliert. Eine weitere Verbesserung der Regel ist nicht erforderlich.

Wenn die Regel für die gewünschte Klasse seltener als ‚Min. Richtig ' zutrifft, d.h. die Regel dem mit Minus (-) markierten Gebiet in Abbildung 2.4 zuzuordnen ist, befriedigt die Regel die Suchbedingungen nicht. Solche Regel wird aus der Arbeit ausgeschlossen und eine weitere Verbesserung der Regel wird nicht durchgeführt.

Die Verfahrensweise für die Regeln, die in den mit Plus und Minus markierten Gebieten liegen, ist offensichtlich. Nicht so offensichtlich ist dies für Regeln, die in der Abbildung 2.4 dem mit einem Fragezeichen (‚?') markierten Quadranten zugeordnet werden. Derartige Regeln können weder ausgeschlossen werden, noch als richtige Regeln protokolliert werden. Das heißt, dass die Regel unvollständige ist. Der Algorithmus hat solche unvollständigen Regeln durch Hinzufügung von Konjunktionen weiter zu vervollständigen bis sie einem der mit Plus oder Minus markierten Bereiche zuzuordnen sind. Für diese Vervollständigung solcher Regeln werden die folgenden (4. und 5.) Spezifikationen verwendet.

4. Kriterium zur **Einschätzung der Qualität der neuen logischen Regel**, die durch Konjunktion mit einem weiteren Merkmal aus der alten, unvollständigen Regel hervorgegangen ist. Dieses Kriterium ist für die gefundenen logischen Regeln, die die Bedingungen 1, 2 und 3 befriedigen, nicht wichtig. Es wird verwendet, um unnütze Konjunktionen zu erkennen und damit zu verwerfen. Sein Sinn ist ähnlich dem *InformationGain* in den Algorithmen der Baumkonstruktion. Aber in den Algorithmen der Baumkonstruktion wird nur ein einziges Merkmal gesucht, das die wirksamste Verzweigung realisiert. Im neuen Algorithmus zur Regelextraktion wird das Kriterium für die Bewertung der Effektivität der Konjunktion von mehreren Merkmalen zu den unvollständigen logischen Regeln verwendet. Vier mögliche Spezifikationen des Kriteriums zur Einschätzung der Qualität der neuen logischen Regeln sollen diskutiert werden:

   Information gain (*IG*) bewertet die neue Regel mit hinzugefügter Konjunktion im Vergleich zur ursprünglichen Regel, das heißt wie durch Ergänzung der Variable $x = Y$ die Regel verbessert wird. *IG* ist für eine Klasse *i* nach der Formel gerechnet:

$$IG(W, x) = I(W) - I(W_{xv}) > 0 \quad (2.4)$$

W ist eine Untermenge der Objekten, die durch die ursprüngliche Konjunktion erkannt wird,

W_{xv} ist eine Untermenge der Objekten, die durch die neue Konjunktion erkannt wird,

$I(W)$ und $I(W_{xv})$ werden nach der Formel berechnet:

$$I(W) = 1 - p(C_i) \quad (2.5)$$

$p(C_i) = |C_i| / |W|$ ist die Wahrscheinlichkeit (verhältnismassige Häufigkeit) der Klasse i im Satz S.

In der hier entwickelten und für Beispiele abgewandten Regelextraktionsmethode wird als eine solche neue, zusätzlich angefügte Konjunktion akzeptiert, wenn die neue Regel Objekte weniger häufig falsch klassifiziert als die alte Regel, wobei die Differenz der falsch klassifizierten mit neuer und alter Regel größer sein muss als der Parameter **Min. Schritt.**

χ^2-Test überprüft die Hypothese der statistischen Verschiedenheit der Erkennungsfähigkeit der neuen und alten Regel (d.h. mit und ohne zusätzliche Konjunktion) [Lane1999]. Dabei wird als zusätzliche Parameter das Konfidenzintervall χ_{max} gefordert. Das Herangehen kann für einige Fälle interessant sein. Das Problem ist aber, dass bei kleinen Objektanzahlen in der Klasse dies Herangehen nicht sinnvoll ist, weil man eine zu große Größe χ_{max} verwenden muss. Bei den nachfolgend untersuchten Arrayanalysen ist dieses Problem gegeben, so dass der **χ^2-Test nicht angewandt wird.**
Die Beschränkung des Konjunktionsranges zur Zurückweisung von zusätzlichen Konjunktionen. Es soll im Folgenden auf eine solche Beschränkung verzichtet werden.

5. Eine Methode oder ein Kriterium für die **Merkmalssortierung** ist erforderlich, weil die Konjunktion mit zusätzlichen Merkmalen in einer definierten Reihenfolge von Variablen erfolgen soll. Auch diese Methode ist nur von Belang für diejenigen Regeln, die die Bedingungen 1, 2 und 3 nicht befriedigen, also in Abbildung 2.4 mit einem Fragezeichen markiert wurden. Die Merkmalssortierung kann erfolgen durch Ranking anhand
   a. eines kombinierten Kriteriums, das richtige und falsche Erkennungen benutzt. Es gibt einige Varianten von diesem Kriterium. In der vorliegenden Arbeit wurde das Kriterium *KK* (kombiniertes Kriterium) getestet:

$$KK = 2\,p(C_i) - 1 \quad (2.6)$$

i ist die Klasse, für die die Suche durchgeführt wird,
$p(C_i) = |C_i| / |W|$ ist die Wahrscheinlichkeit (verhältnismassige Häufigkeit) der Klasse i im Satz W.

   b. sowie anhand der Anzahl falsch klassifizierter Objekte.

Bei der nachfolgend diskutierten Anwendung des neuen Algorithmus zur Regelextraktion wurde gefunden, dass die Ergebnisse für die beiden Methoden der Sortierung (a und b) nahezu identisch sind. Die Methode a) arbeitet etwas langsamer, deshalb wurde bei der Durchführung aller nachfolgend dargestellten Untersuchungen die zweite Methode (b) der Sortierung verwendet.

Ein neue Algorithmus zur Regelextraktion ist in Abbildung 2.5 dargestellt. Eine wichtige Charakteristik des Algorithmus ist die Geschwindigkeit und die Fähigkeit, die großen Datenmengen zu bearbeiten. Der Algorithmus angewandt auf eine künstlich erzeugte Matrix von 100 Objekten, 10000 Variablen (Ge-

nen) und 10 Diskretisierungsstufen benötigte einige Stunden Rechenzeit auf einem gewöhnlichen Personalcomputer (AMD 1700; 2Gb). Es wurden einige Hunderttausend Regeln erzeugt und aus diesen im Laufe einer Stunde die Reduktion des Regelsatzes zur ‚Disjunktiven Normalform' (*DNF*) durchgeführt. Es ist offensichtlich, dass die erforderliche Rechenzeit von den gewählten Parametern abhängt. Die Änderung eines des ersten drei Parameter kann die Arbeit des Programms beschleunigen oder verzögern. Aus Gründen der Begrenzung der Rechenzeit auf einige Stunden soll für das Anwendungsgebiet der Genexpressionsdatenanalyse die Anzahl der selektierten und verwendeten Merkmale (Gene) einige Dutzend nicht überschreiten. Die Bearbeitung der von solchen Matrizen mit den Parametern Min. Richtig =3 und Max. Fehler = 0 benötigt eine Rechenzeit von Millisekunden bis zu einigen Sekunden. Auch die *DNF*-Erzeugung dauert bis zu einigen Sekunden.

In der Tabelle 2.1 sind die Hauptcharakteristiken des neu entwickelten Algorithmus im Vergleich zu den Algorithmen der Entscheidungsbaumkonstruktion und dem Algorithmus *Kora* dargestellt. Man kann leicht bemerken, dass der neu entwickelte Algorithmus von den existierenden Algorithmen wesentlich verschieden ist.

Wo	Was	Entscheidungsbaum	*Kora*	Neue Methodik
Input	Konjunktionsrang	-	+	Optional
	min. Richtig	-	+	+
	max. Fehler	-	+	+
	minimale Anzahl der Objekten in Eltern (Kinder) Knoten	+	-	-
Verzweigung	InformationGain / Qualität	+	-	Optional
Stopkriterium	Regel „minimale Anzahl der Objekten in Eltern (Kinder) Knoten"	Optional	-	-
	Alle Objekten sind erkannt	Optional	-	-
	Es gibt keine wichtigen Variablen für die Verzweigung	+	-	+
	Es gibt keine Variablen für die Verzweigung	+	+	+
Pruning	max. Korrelation für 2 Regeln	Man kann nutzen	Optional	-
	Max. Merkmalszahl in Konjunktionen	Man kann nutzen	Optional	-
	DNF- Erzeugung	-	Man kann nutzen	+
	cross – Validation	Optional	-	-
Algorithmus	Regelextraktion pro eine Durchgang für	> 1 Klasse		1 Klasse

Tabelle 2.1. Vergleichanalyse der Methoden

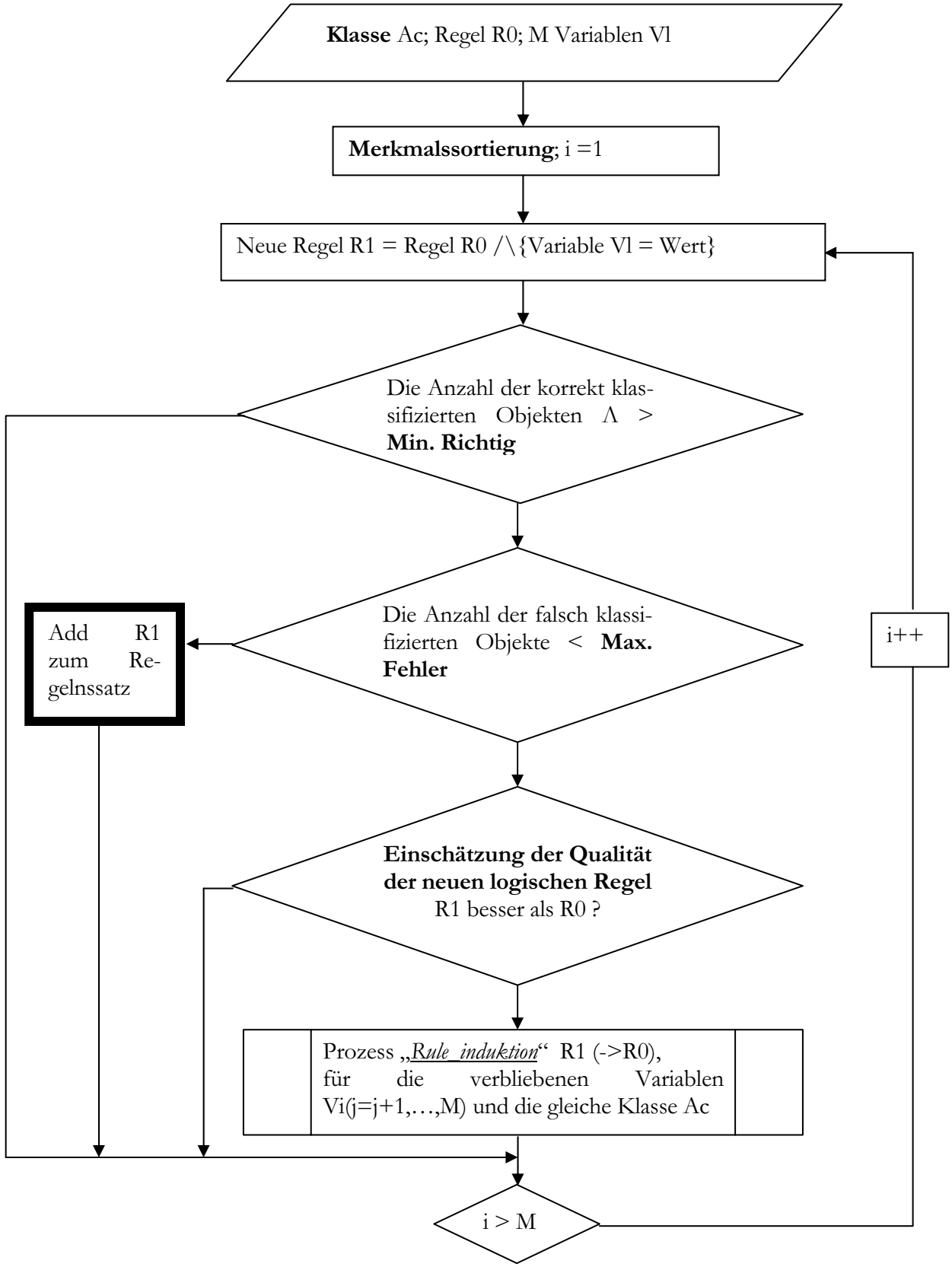

Abbildung 2.5 Schema des Regelextraktionsalgorithmus (GeneRule 2.0). Prozess „Rule_induction“ wird mit einer Liste von Variablen Vi und der leeren Regel R0, gestartet.

2.2.3 Regelreduktion

Bei der Erzeugung der logischen Regeln durch die oben untersuchte Methodik wird oft eine große Anzahl von Regeln geschaffen. Weil von den vielen logischen Regeln für das Objekt nur ein kleiner Teil abstimmen (votieren, „feuern") wird, ist bei der Erarbeitung der „*decision making*" Methodik die Anzahl der logischen Regeln unwichtig . Wenn die verwendeten logischen Regeln richtig sind, ist eine Reduktion der Regelmenge nicht erforderlich.

Die Verkleinerung der Menge der logischen Regeln ist nützlich zur Verbesserung der Übersichtlichkeit und Transparenz (Elimination redundanter Regeln) und notwendig zur Elimination falscher Regeln. Bei der Verringerung der Zahl der logischen Regeln besteht die Gefahr, auch richtige Regeln zu verlieren, die für das bessere Verstehen der betrachteten Probleme interessant sind oder für die korrekte Klassifikation erforderlich sind. Solche Probleme treten auch bei der Entscheidungsbaumkonstruktion auf. Ziel der Regelreduktion ist also nicht die Minimierung der Regelmenge sondern deren Optimierung.

Wichtig ist dabei die Einschätzung der Richtigkeit der Regeln, also deren Validierung.

Die benutzten Daten sind nicht universell, d.h. sie decken nicht den gesamten Zustandsraum ab, und ihre Struktur (die einzelne Objekte mit Tausenden Merkmalen) kann zu falschen oder zweifelhaften Regeln führen. Komplizierte Regeln, d.h. solche mit mehreren Konjunktionen (Konjunktionsrange >1) machen weniger Fehler des Typs 1 (falsch Positive) und mehr Fehler des Typs 2 (falsch Negative). Die erfolgreiche Kombination der komplizierten Regeln kann zur vollständigen und fehlerfreien Erkennung führen. Methoden der Regeloptimierung werden selten in der Literatur betrachtet. Wir werden nachfolgend einige mögliche Methoden der Reduktion von Regelmengen diskutiert.

1. Erzeugung der Disjunktiven Normalform (*DNF*)

   Die Methoden zur *DNF*-Erzeugung basieren auf der diskreten Mathematik und der Algebra der Logik. Sie sind gut bekannt und beschrieben (s. z.B. [Aigner1996] und 31.2). Es ist offensichtlich, dass diese Methoden universell gültig und richtig sind. Der Test des *DNF*-Generators hat gezeigt, dass er in akzeptabler Zeit fähig ist, die Bearbeitung der einigen Hundert Tausenden von logischen Regeln zu leisten. Diese Methodik der *DNF*-Erzeugung wird nachfolgend bei der Analyse aller Daten genutzt.

2. Clusteranalyse

   In diesem Fall wird jede Konjunktion als ein separates Merkmal betrachtet und es wird der Raum der logischen Regeln anstelle des Raumes der Merkmale analysiert. Die Clusteranalyse im Raum der logischen Regeln ermöglicht die Erkennung der besonderen Regeln und die Bewertung sowie mögliche Aggregation von logischen Regeln. Diese Methodik ergibt interessante Ergebnisse, die zum Verständnis des Problems beitragen. Bei großer Anzahl von logischen Regeln ist dieses Herangehen jedoch ziemlich kompliziert und der Wert die erhaltenen Ergebnisse ist zweifelhaft. Im Rahmen der vorliegenden Arbeit wurden zwei Programme zur Clusteranalyse im Raum der logischen Regeln geschrieben:

   a. Das Programm der automatische Clusteranalyse (*unsupervised learning*):. Ein unbefriedigend gelöstes Problem ist die optimale Wahl der Clusteranzahl. Dies Problem tritt ähnlich auch bei Baumkonstruktionsalgorithmen auf. Mit diesem automatischen Herangehen war es nicht möglich, stabil gute Ergebnisse zu bekommen. Deshalb werden hier die Ergebnisse nicht präsentiert.
   b. Das Programm der Objekterkennung: Das Programm leistet die direkte Objektenerkennung. Die Analyse wird manuell unterstützt. Die Nutzung ist bei kleiner Anzahl von logischen Regeln wirksam. In der nachfolgenden Datenanalyse für ausgewählten Beispiele wird dies Programm oft verwendet und nachfolgend für die Beispiele „Gesichter" und „Hirntumor" benutzt.

3. Regelvalidierung aufgrund biomedizinischer Schlüsselwörter und logischer Tests.

   Die logischen Regeln kann man als einfache Konjunktionen (Vereinigung) von Merkmalen (z.B. Genen oder Proteinen) betrachten, die mit medizinisch-biologische Begriffen (Schlüsselwörter) verbunden sind. Für die Realisierung dieser Methode wurden im Rahmen dieser Arbeit einige Programme geschrieben, die der Teil des Paketes *GeneRule* Version 2.0 sind. Diese Methodik wird nachfolgend für die Analyse aller drei in der Arbeit studierten biologischen Beispiele verwendet. In allen

drei Fällen führte die Anwendung dieser Methode zu einer erheblichen Reduktion der Menge der logischen Regeln. In zwei der drei Fälle führte die Anwendung der Methode zu besserer Qualität des Erkennens.

4. Die Methodik, die auf der Experteneinschätzung der logischen Regeln gegründet ist.

   In diesem Fall bestimmt ein Experte (der Spezialist im vorliegenden Gebiet) selbständig die Richtigkeit der logischen Regeln und löscht oder korrigiert sie falls notwendig. Die vorliegende Methodik ist wahrscheinlich die nützlichste und die wichtigste, aber sie war für die vorliegende Arbeit leider nicht anwendbar, weil das erforderliche Fachwissen nicht zur Verfügung stand.

2.2.4 Decision-making Methode

Bei der Anwendung und der Beurteilung der Güte von Erkennungsalgorithmen besteht die Aufgabe, die erzeugten Regeln für die Klassifikation (Erkennung) zu nutzen. Die „Abstimmung" der Merkmale (das *voting*) ist eine Methode bei der die Fehlerwahrscheinlichkeit bei der Klassifikation gesenkt wird.

Das Wesen der „Abstimmung" - Methode besteht darin, dass für das Objekt k (ein Subjekt der Klassifikation), in jeder Klasse A_i die „Stimmen" $n_i(k)$ für diese Klasse gezählt werden. Das Objekt wird jener Klasse zugeordnet, für die die meisten „Stimmen" gezählt wurden.

In [Poljakova1976] sind die Möglichkeiten der Nutzung der „Abstimmung", für die Verbesserung der Erkennungsergebnisse untersucht wurden.

Aus der Menge der Objekte A ist das wahrscheinliche Maß $\gamma(B_k)$ angegeben, das aus allen Teilmengen $C_k \subset A$ bestimmt wird. Für das bessere Verständnis wird der Fall untersucht, dass die Menge der Objekte nur aus zwei Klassen A_1 и A_2 besteht.

Mit q_i ($\Delta n < 0$) wird die bedingte Wahrscheinlichkeit bezeichnet, dass für ein Objekt $k \in A$ die Differenz $\Delta n = n_i(k) - n_j(k)$ der Zahl der „Stimmen" für diese und die entgegengesetzte Klasse negativ wird. Die Wahrscheinlichkeit des Fehlers bei der Klassifikation eines Objektes $k \in A$ durch die Methode der „Abstimmung" ist

$$q = \gamma(A_1) q_1(\Delta n < 0) + \gamma(A_2) q_2(\Delta n < 0), \qquad (2.7)$$

$\gamma(A_i)$ ist die Wahrscheinlichkeit, dass das Objekt k zur Klasse A_i gehören wird.

$M_i(\Delta n)$ ist der Erwartungswert der Differenz Δn der Stimmenzahl für „seine" und die „fremde" Klasse. Unter der Bedingung, dass das Objekt zur Klasse A_i gehört und $D_i(\Delta n)$ die Dispersion dieser Differenz bei derselben Bedingung ist und $M_i(\Delta n) > 0$ ist, kann entsprechend die Chebishev -Ungleichheit aufgezeigt werden, dass folgende Ungleichung gilt.,

$$q_i(\Delta n < 0) <= D_i(\Delta n) / (M_i(\Delta n))^2 \qquad (2.8)$$

Bei der positiven Bedingung $M_i(\Delta n)$ gilt:

$$M_1(\Delta n) = M_1(n_1(k)) - M_2(n_2(k)) \text{ und } M_2(\Delta n) = M_2(n_2(k)) - M_2(n_1(k)) \qquad (2.9)$$

$M_j(n_i(k))$ ist der Erwartungswert der Zahl der „Stimmen" für A_i unter der Bedingung, dass das Objekt k zur Klasse A_i gehört:

$$M_j(n_i(k)) = \sum_{\beta_i^l \succ A_i} \gamma_i(\beta_i^l) \qquad (2.10)$$

$\gamma_i(\beta_i^l)$ ist das Maß der Kreuzung der Klasse A_i auf dem Gebiet der Richtigkeit der Regel β_i^l, die für die Klasse A_i extrahiert wird. $\beta_i^l \succ A_i$ bedeutet, dass Alle gültigen Regeln β_i^l der Klasse A_i summiert werden.
Wenn für jede Klasse nur die genügend sicheren Regeln verwendet werden, so wird die Fehlerwahrscheinlichkeit für jede Klasse kleiner als 1/2. Dann gilt für alle Regeln β_1^l der Klasse A_1.

$$\gamma_1(\beta_1^l) > \gamma_2(\beta_2^l) \tag{2.11}$$

Für alle Regeln β_2^l der Klasse A_2 gilt

$$\gamma_2(\beta_2^l) > \gamma_1(\beta_1^l) \tag{2.12}$$

Aus (2.10) - (2.12) folgt: $M_1(n_1(k)) > M_2(n_1(k))$, $M_2(n_2(k)) > M_1(n_2(k))$. (2.13)

Für die Erfüllung der Bedingung der gleichzeitigen Positivität von $M_1(\Delta n)$ und $M_2(\Delta n)$ ist es hinreichend zu fordern die Befriedigung der Gleichung 2.13 (s. 2.10) :

$$M_1(n_1(k)) = M_2(n_2(k)) \tag{2.14}$$

Aufgrund der untersuchten Bedingungen wurden in derselben Arbeit die folgenden Empfehlungen für die Merkmalsauswahl zur Anwendung der Abstimmungs-Methode gegeben. Da die Fehlerwahrscheinlichkeit q bei der Klassifikation und die mittlere Bedeutung der Wahrscheinlichkeit q_i ($\Delta n < 0$) gleich sind, so ist es für die Verkleinerung der Wahrscheinlichkeit q notwendig:

1) die Bedingung (2.14) befolgend, die Regeln mit der kleinsten Wahrscheinlichkeit des Fehlers anzunehmen.

2) zur Verringerung der Dispersionen $D_i(\Delta n)$ werden nur solche Regeln angenommen, dass alle Objekte der Klasse A_i ungefähr auf die Regeln gleich verteilt sind.

Die gleichen Autoren haben für die Verbesserung des Algorithmus vorgeschlagen, jeder der Regeln β_i^l ein Gewicht $r(\beta_i^l)$ zuzuschreiben und die Zählung der Zahl der „Stimmen“ durch die Zählung der Summe der Gewichte zu ersetzen. Die Gleichung für den Erwartungswert der Summe $\tau_i(k)$ des Gewichtes ist

$$M_j(\tau_i(k)) = \sum_{\beta_i^l \succ A_i} \gamma_i(\beta_i^l)\, r(\beta_i^l) \tag{2.15}$$

Es ist offensichtlich, dass die Zählung der „Stimmen“ und die Zählung im Fall der Zuordnung der Regeln zu den einzelnen Gewichten „1“ äquivalent ist.

Für die Korrektur der bekommenen Ergebnisse werden in der vorliegenden Arbeit die Gewichtskoeffizienten verwendet. Als Gewichte der Regeln für die Klasse A_i wird die Größe $1/m_i$ (m_i ist die Anzahl der Regeln für die Klasse A_i.) verwendet. Der Erwartungswert der Summe der Gewichteskoeffizienten $\tau_i(k)$ ist:

$$M_j(\tau_i(k)) = \frac{1}{m_i} \sum_{\beta_i^l \succ A_i} \gamma_i(\beta_i^l). \tag{2.16}$$

Das Wesen der „Abstimmung“ besteht in diesem Fall darin, dass für das Objekt k (ein Objekt der Klassifikation) in jeder Klasse A_i die Zählung der Gewichtssumme $n_i(k)/\ m_i$ der „Stimmen“ für diese Klasse erfolgt. Das Objekt k wird zu jener Klassen bezogen, für die die Gewichtessumme am größten ist.

2.2.5 Ein Beispiel der Regelanalysis

Als Demonstrationsbeispiel für die Erzeugung von logischen Regeln wird der Datensatz „Männergesichter“ aus [Djuck2001] verwendet. Die Aufgabe ist intuitiv klar. Die Vorteile des Datensatzes sind seine Anschaulichkeit und die leichte Überprüfbarkeit der erhaltenen Ergebnisse.

Klasse	Regeln	Erklärung der Regeln
1	$X_6 = 1$ und $X_{11} = 1$ und $X_{16} = 1$	Naselippen Falte und Brille und Pfeife
	$X_{10} = 1$ und $X_{15} = 1$ und $X_4 = 1$	Bart und Ohrring und runde Augen
2	$X_2 = 1$ und $X_7 = 1$ und $X_{13} = 1$	Angedruckte Ohren und dicken Lippen und Schmetterling
	$X_3 = 1$ und $X_{14} = 1$ und $X_9 = 1$	Runde Nase und Augenbrauen nach oben und Schnurrbärte

Tabelle 2.2 Erhaltene Logische Regeln (End Version)

Die erhaltene logische Regelbasis (s. Tab. 2.2 und Anhang B) wird im nachfolgenden Abschnitt mit zwei anderen Klassifikationsmethoden verglichen. Durch die Anwendung des Diskriminators aus dem bekannten Softwareprogramm STATGRAPHICS erhält man die nachfolgende Funktion:

$$F = -40,1 - 11,0x_1 + 23,7x_2 + 6,8x_4 - 7,0x_5 - 6,3x_6 + 24,2x_7 - 15,0x_8 - 6,4x_{11} + 34,\ 3x_{12} + 14,1x_{13}$$

De Formel erlaubt die Unterscheidung der Objekte zwischen Klasse 1und 2. Der Nachteil dieses Klassifikators ist, das man keine neuen Informationen über die Objekte des Datensatzes gewinnen kann. In der Abbildung 10.3 ist ein Entscheidungsbaum, der mit Hilfe des *ID3* Algorithmus erzeugt wurde, dargestellt.

Die Anzahl, der aus dem Baum ableitbaren Regeln, ist mit 7 größer als die mit *DataControl* erhalten Regelbasis. Ein weiterer Nachteil des Entscheidungsbaumes ist, dass nicht alle Objekte korrekt klassifiziert werden.

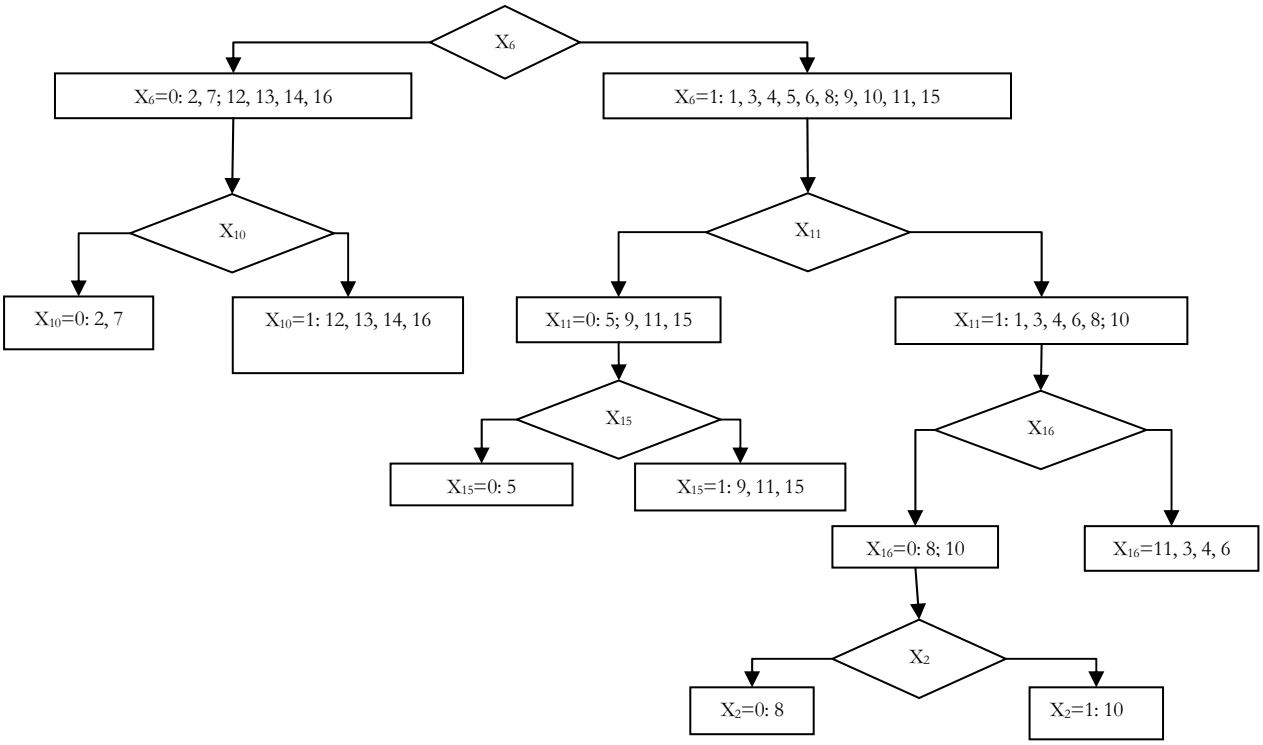

Abbildung 10.2 Entscheidungsbaum für „Gesichter“ (*ID3*)

2.3. Fazit

1. Die wirksame Diskretisierung und die Variablenauswahl sind durch das statistische Herangehen möglich. Damit ist eine Datennormalisierung nicht erforderlich.

2. Die neue Methode zur Erzeugung von logischen Regeln besteht in der Kombination zweier Methodiken, nämlich der Entscheidungsbaumkonstruktion und *Kora*. Die neue Methode unterscheidet sich wesentlich von diesen beiden Methoden.

3. Eine Reduktion der Regelmengen und Konjunktionen in den Regeln, die durch die neue Methode zur Regelgenerierung erhalten werden, ist bei der Genexpressionsanalyse durch die folgenden Methoden möglich:
   - die Nutzung der Regeln der diskreten Mathematik und Algebra der Logik (*DNF*-Erzeugung);
   - Clusteranalyse;
   - das logische Tests über Schlüsselwörter (s. nachfolgendes Kapitel 3).
4. Die Schlussfolgerung aus der Regelmenge wird durch Abstimmung über die Gewichtssumme getroffen.

3 Validation der regelbasierten Wissen

3.1. Regelanalysemethode

3.1.1. Theoretische Grundlagen

Am häufigsten suchen oder erarbeiten die Forscher auf dem Gebiet der Genexpressionsanalyse eine Methodik für die mathematische Datenanalyse. Die fachspezifische Analyse der Ergebnisse wird häufig nicht durchgeführt und ist vielfach nicht möglich (siehe die Ergebnisse der Korrelationsanalyse für «Gesichter»). Logische Regeln sind jedoch für eine fachspezifischen Analyse geeignet. Anderseits, ist es bekannt, dass eine geringe statistische Signifikanz nicht zwangsläufig bedeutet, dass die Regel biologisch uninteressant ist [Clare2002].

Nachfolgend sollen Möglichkeiten der Nutzung logischer Tests in Verbindung mit Fachbegriffen (Schlüsselwörtern) diskutiert werden.

Die vorliegende Arbeit ist konzentriert auf die Genexpressionsdatenanalyse mit dem Ziel der Diagnostik „gesund" – „krank", oder „die Erkrankung 1"-„die Erkrankung 2".

*	Erste Krankheit	Zweite Krankheit
1	manchmal/immer	niemals
2	manchmal/immer	manchmal/immer
3	niemals	niemals
4	niemals	manchmal/immer

Tabelle 3.1. Merkmalskombinationen für zwei Krankheiten.

Für jede der Erkrankungen kann man die Symptome auf drei Erscheinungsformen [Monossov1994] aufteilen:

- die erscheinen immer oder manchmal
- die erscheinen niemals.

Diese Erscheinungsformen der Symptome sollen nachfolgend verwendet werden. So sind für zwei Erkrankungen die in Tabelle 3.1. genannten Kombinationen der Merkmale möglich. Die Kombinationen 1 und 4 sind besonders interessant, weil sie eine Unterscheidung von Krankheitszuständen erlauben, während die Kombinationen 2 und 3 keine Unterscheidung von Krankheitszuständen erlauben und deshalb als nutzlos von der weiteren Betrachtung ausgeschlossen werden.

In der ersten Etappe werden logischen Regeln generiert, wie im Kapitel 2 beschrieben wurde. Auf dem Lerndatensatz kann jede der erhaltenen logischen Regeln einige der Objekte richtig erkennen. Da häufig der Lerndatensatz nur einen geringen Teil von allen möglichen Objekten umfasst, ist es möglich, dass die erhaltenen Regeln zufällig oder falsch sind. Diese logischen Regeln sind statistisch signifikant nur für den gegebenen Lerndatensatz. Jede Regel trägt konkrete Informationen, die Zusammenhänge in den untersuchten Prozessen (z.B. Erkrankungen) widerspiegeln. Diese Informationen beziehen sich auf die Beschreibung von Merkmalen, in unserem Falle der Expression von Genen (Proteinen). Wenn die Merkmale (Gene) durch die für sie charakteristischen Schlüsselwörter ersetzt werden, können diese so transformierten Regeln zur Validation der Regeln verwendet werden (Abbildung 3.1).

Die Begriffe (Symptome, Schlüsselwörter) können als richtig (zur Unterscheidung geeignet) gelten, wenn die nur in einem der beiden zu unterscheidenden Fälle zutreffen (Fall 1 für die erste Krankheit oder Fall 4 für die zweite Krankheit in der Tabelle 3.1). Andererseits, wenn ein Schlüsselwort bei beiden Zu-

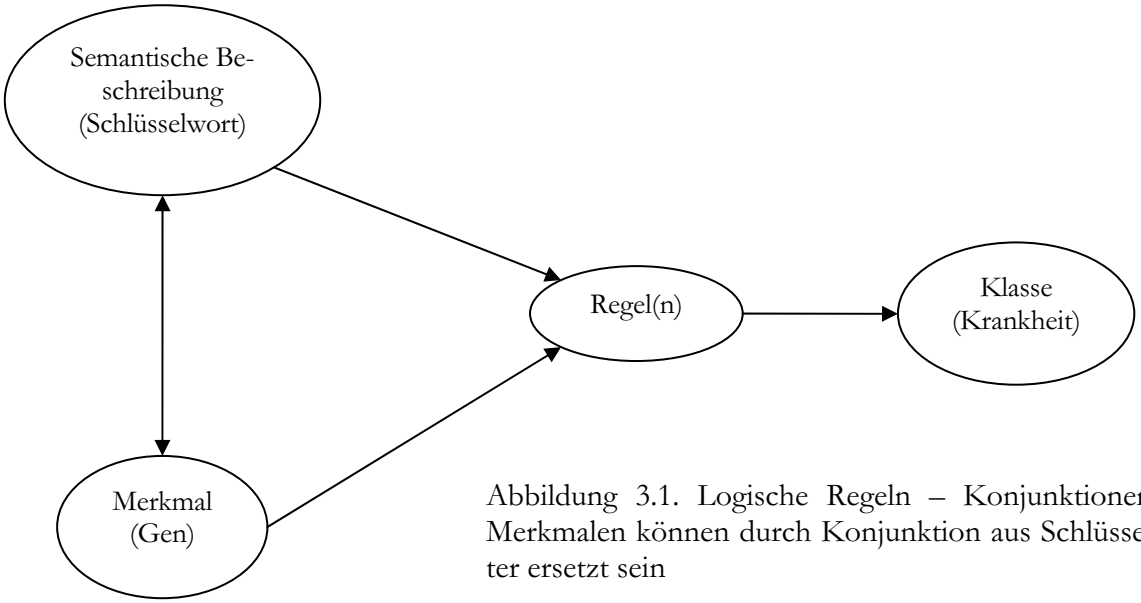

Abbildung 3.1. Logische Regeln – Konjunktionen aus Merkmalen können durch Konjunktion aus Schlüsselwörter ersetzt sein

ständen identisch zutrifft, so weist dies auf einen möglichen Widerspruch hin. Die Regel kann dann als falsch (Fälle 3 und 4 für die erste Krankheit oder Fälle 1 und 3 für die zweite Krankheit in der Tabelle 3.1) oder zufällig (Fall 2 in der Tabelle 3.1) verworfen werden.

3.1.2. Praktische Regelanalyse

Die Merkmale werden in den Regeln durch Schlüsselwörter ersetzt. Dann gilt eine Regel als gültig (validiert), wenn folgende Bedingungen erfüllt sind:

- Zu keiner Konjunktion dürfen sich gegenseitig ausschließenden Merkmale gehören;
- Regeln, die für dieselbe Klasse zutreffen, dürfen keine sich gegenseitig ausschließenden Merkmale haben;
- Regeln, die für unterschiedliche Klassen zutreffen, dürfen nicht identische Merkmale haben

Die diese drei Bedingungen nicht befriedigenden Konjunktionen und Schlüsselwörter sind wahrscheinlich falsch. Sie werden aus der Regelmenge entfernt, wenn ihre Entfernung die Güte der Erkennung nicht schadet.

	Beschreibung	NUCLEAR	NUCLEAR PROTEIN	MONOCYTE
1. Regel	IF 3[h] and 6[l] THEN ALL			L
2. Regel	IF 4[h] and 6[l] THEN ALL	H	H	L

Abbildung 3.2. Ein Beispiel der Regelreduktion. Die Beschreibung im Text.

Nach Anwendung dieser auf Schlüsselwörtern basierenden Reduktion der Regelmenge erfolgt wieder die Reduktion mittels der *DNF*-Erzeugung nach den Regeln der diskreten Mathematik. Ein Beispiel für das Löschen ist in Abbildung 3.2. dargestellt. In diesem Beispiel gibt es zwei Regeln mit je zwei Merkmalen. Die zweite Regel kann gelöscht werden, wenn das Löschen der zweiten Regel das Erkennen nicht verschlechtert.

Die hier vorgeschlagene Methode ist ähnlich zu der in [Oyama2002] verwendeten Methode (Abbildung 3.3 und 3.4). Die Autoren [Oyama2002] nutzten die Beschreibungen (*Keywords*) der Proteine für die Analyse und die Verallgemeinerung der Reaktionen und Wechselwirkungen, an denen die Proteine teilnehmen. Für die Durchführung dieser Arbeit [Oyama2002] wurden die Daten aus biologischen Datenbanken für Protein-Protein-Wechselwirkungen und die Beschreibungen von Proteinen verwendet (Abbildung 3.3). Die Proteine wurden durch ihre Beschreibungen (*Keywords*) ersetzt. Daraufhin wurden die Reaktionen und Wechselwirkungen zwischen den Proteinen durch die Reaktionen und Wechselwirkungen zwischen den Merkmalen ersetzt (Abbildung 3.4). Auf diese Weise haben die Autoren vollkommen neue Information über die Reaktionen und Wechselwirkungen erhalten. Das in der vorliegenden Arbeit verwendete Herangehen ist ähnlich. Es unterscheidet sich jedoch insbesondere in der Generierung der Schlüsselwörter.

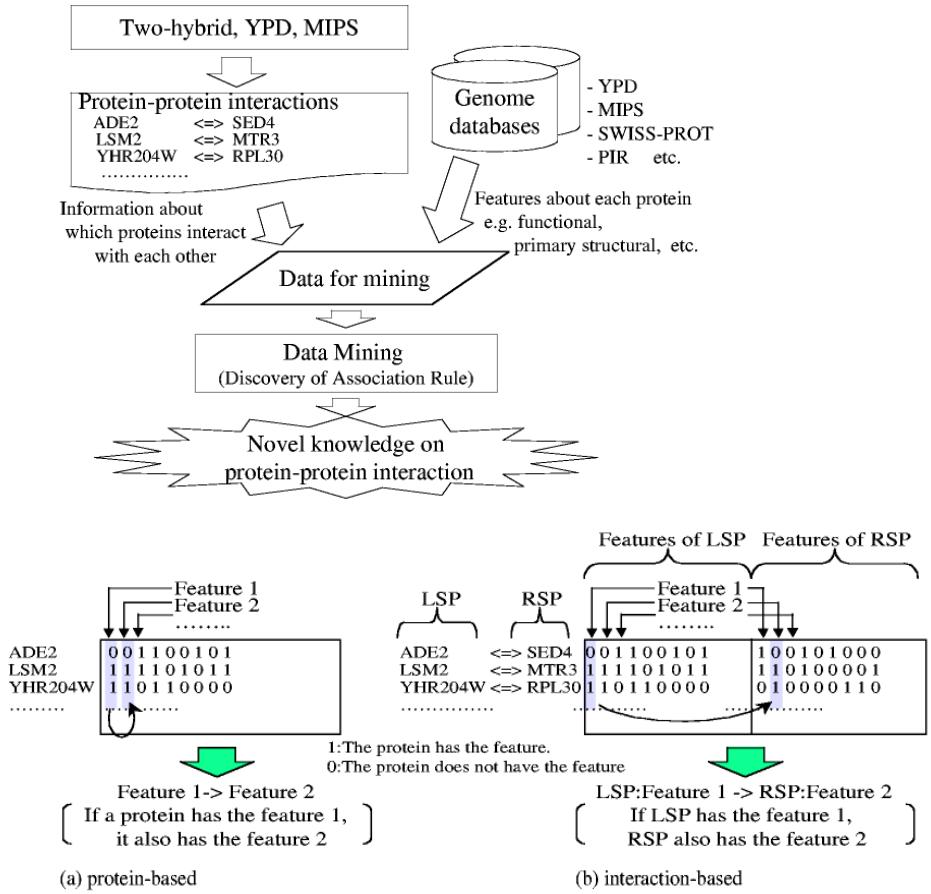

Abbildung 3.3 (oben) and 3.4 (unten). Aus: [Oyama2002]. Die Daten sind aus Tausenden Protein-Protein-Wechselwirkungen und Schlüsselwörtern für die Proteine Merkmale aufgebaut. Dann wird neues Wissen über die Protein-Protein-Wechselwirkungen durch *Data Mining* als logische Regeln extrahiert

Die im folgenden Kapitel 3.1.3 dargelegt Methode zur Schlüsselwortextraktion beruht auf folgenden Überlegungen:

- Eines der ernsten Probleme, die bei der Analyse der logischen Regeln besteht, ist die verständliche Darstellung der Ergebnisse. Eine Regel in der Form «Das Gen X ist bei der Erkrankung Y exprimiert» bleibt oft unverständlich. Eine verständlichere Form wäre mit einer Erklärungskomponente verbunden: «Das Gen X, das die Funktion F hat und gewöhnlich auch bei den Erkrankungen C und B exprimiert,

ist bei der Erkrankung Y exprimiert » . Aber, um solche Erklärungskomponente zu erhalten, ist es erforderlich, die Information über das Gen in speziellen Datenbanken durchzusehen und die wichtige Information von der großen Menge der unnützen Fakten zu extrahieren. Bei Dutzenden und mehr Genen, die in der Regelmenge vorkommen, ist diese Aufgabe manuell nicht zu leisten. Die automatische Transformation der Regeln in die Form, die für das Verständnis bequem ist, ist deshalb eine wichtige Aufgabe.

Jede Regel besteht aus einer Konjunktion von Merkmalen. Jede Regel gehört nur zu einer Klasse. Jedes der Merkmale kann durch andere Merkmale beschrieben sein. In unserem Fall sind die Merkmale die Expressionshöhen der Gene, für die standardisierten Beschreibungen existieren. Wenn die Gene durch diese standardisierten Beschreibungen ersetzt werden, können diese Regeln interpretiert und damit kann das Verständnis der Regeln bedeutend verbessert werden.

- Beim Ersetzen der Gene durch ihre Beschreibungen können neue Informationen erhalten werden. Ein Urteil über die Richtigkeit und Qualität der neuen Informationen kann jedoch nur der Fachexperte treffen. In vielen Fällen sind jedoch die konkreten Mechanismen der Erkrankung auf dem biochemischen Niveau ziemlich kompliziert und nicht klar. Dann sind die erhaltenen neuen Informationen lediglich Hypothesen, die der weiteren Forschung als Grundlage dienen können.
- Wie noch gezeigt werden wird, kann man einerseits die Menge der Regeln verringern und gleichzeitig die Klassifikatioinsgüte verbessern.

3.1.3. Schlüsselwörterextraktionstechnik

In [Oyama2002] wurden funktionale, strukturelle und andere Merkmale verwendet, die man auf sieben Typen einstufen kann.

- *YPD Kategorien:* 275 Merkmale.
- *EC Nummer:* 154 Merkmale
- *SWISS-PROT/PIR* Schlüsselwörter: 491 Merkmale
- *PROSITE motifs:* 698 Merkmale
- *Bias von Aminosäuren:* 20 Merkmale
- *Segmentcluster:* 3589 Merkmale
- *Aminosäuren Pattern:* 14 Merkmale

Die Autoren [Oyama2002] verwendeten nur jene Beschreibungen der Merkmale, die leicht verschlüsselt und bearbeiten werden können. Die Beschreibungen wurden aus verschiedenen Quellen genommen, aber man könnte sie durch die Beschreibungen nur aus *UniProt/Swiss-Prot/TrEMBL* (die Felder CC – *Comment lines*, KW - *Keywords* und FT - *Feature table*) nach ihrer entsprechenden Bearbeitung ersetzen, weil sie alle diese und viele andere Daten enthalten. Die Mängel des Herangehens können durch die Analyse der *UniProt/Swiss-Prot/TrEMBL* – Proteinenbeschreibungen gezeigt werden (Abbildung 3.5). Die Autoren von *UniProt/Swiss-Prot/TrEMBL* wollen offensichtlich nur die wichtigsten Merkmale des Proteins, die dieses Protein von anderen Proteinen unterscheiden, wählen. Solche Parameter sind im Feld „*Keywords line*“ (KW). Nachfolgend sollen wichtige Beschreibungen innerhalb der Datenbank *UniProt/Swiss-Prot/TrEMBL* diskutiert werden.

AC ***P27449***;
DE ***Vacuolar ATP*** synthase 16 kDa ***proteolipid*** subunit (***EC 3***6.3.14).
GN ***ATP6V0C*** OR ***ATP6L*** OR ***ATP6C*** OR ***ATPL***.
CC -!- FUNCTION: ***PROTON-CONDUCTING PORE FORMING*** SUBUNIT OF THE ***MEMBRANE***
CC ***INTEGRAL*** V0 COMPLEX OF ***VACUOLAR ATPASE***. V-***ATPASE*** IS RESPONSIBLE
CC FOR ***ACIDIFYING*** A VARIETY OF ***INTRACELLULAR COMPARTMENTS*** IN
CC EUKARYOTIC ***CELLS***.
CC -!- CATALYTIC ACTIVITY: ***ATP*** + H(2)O + H(+)(In) = ***ADP*** + ***phosphate*** +H(+)(Out).
CC -!- SUBUNIT: V-***ATPASE*** IS AN ***HETEROMULTIMERIC ENZYME*** COMPOSED OF A
CC ***PERIPHERAL CATALYTIC*** V1 COMPLEX (MAIN COMPONENTS: SUBUNITS A, B,
CC C, D, E, AND F) ATTACHED TO AN ***INTEGRAL MEMBRANE*** V0 ***PROTON PORE***
CC COMPLEX (MAIN COMPONENT: THE ***PROTEOLIPID*** PROTEIN; WHICH IS PRESENT
CC AS A HEXAMER THAT FORMS THE ***PROTON-CONDUCTING PORE***).
CC -!- SUBCELLULAR LOCATION: ***Integral membrane*** protein. ***Vacuolar***.
CC -!- MISCELLANEOUS: THIS SUBUNIT BINDS ***DICYCLOHEXYLCARBODIIMIDE*** (***DCDD***)
CC WHICH INHIBITS THE ***ATPASE***.
CC -!- SIMILARITY: BELONGS TO THE V-***ATPASE PROTEOLIPID*** SUBUNIT FAMILY.
KW ***Hydrolase***; ***Hydrogen ion transport***; ***ATP synthesis***; ***Transmembrane***.
FT BINDING 139 139 ***DICYCLOHEXYLCARBODIIMIDE*** (POTENTIAL).

Abbildung 3.5. Ein Teil der Proteinbeschreibung aus *UniProt/Swiss-Prot/TrEMBL*. Zeile KW enthält Schlüsselwörter für dieses Protein. Die markierten Schlüsselwörter sind durch den *Natural Language Processor* (*NLP*) von *GeneRule* (Version 2.0) extrahiert worden.

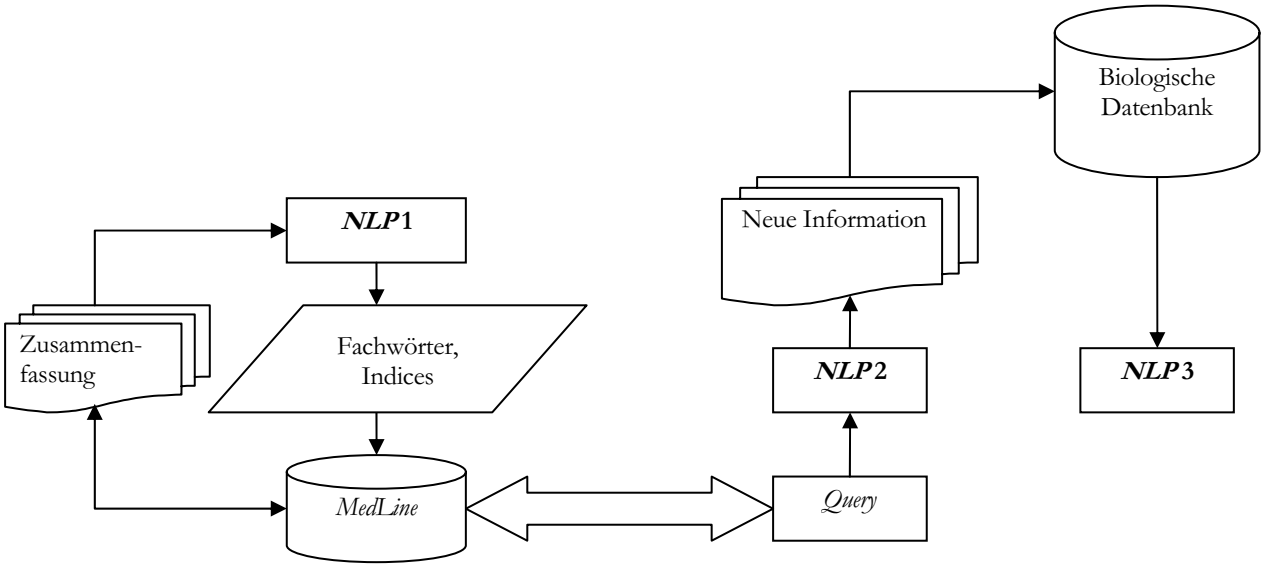

Abbildung 3.6. *NLP* in den medizinischen und biologischen Forschungen.

Molekulargenetische und biochemische Parameter, die im Feld *Feature Tables* (FT) abgelegt sind, wurden verworfen. Stattdessen wurden Textinformationen, die Makroerscheinungen beschreiben, ausgewählt.

Ein großer Teil der Information in den biologischen Datenbasen wird in nicht verschlüsselter Form dargeboten. Solches Format kann man als Textformat verarbeiten. Für die Nutzung der Text-Information ist ein *NLP* erforderlich. Die Literatursuche in PubMed mit dem Schlüssel "*TEXT MINING*" gab 56 Hinweise (im April 2004). Die Analyse dieser Literatur gibt die Tabelle 3.2 und Abb. 3.6. Eine große Menge von *NLP* sind für die Analyse und die Extraktion der Schlüsselwörter aus Texten entwickelt worden. Dabei werden alle wichtigen Fachwörter herausgezogen, die für das Verständnis und die weitere Nutzung wichtig sind. Nur unwichtige Wörter werden verworfen. Solche *NLP* werden breit, zum Beispiel

in PubMed, und auch in beliebigen Suchsystemen des Internets, verwendet. In Tabelle 3.2 werden diese als *NLP*1 bezeichnet. Solche *NLP*1 sind ungeeignet für die Regelvalidierung, weil sie aus den Texten zuviel unnütze Fachwörter extrahieren. So sind die Wörter "Protein", "Gene", Enzyme" für die Regelvalidierung ungeeignet, aber sie werden durch *NLP*1 aus dem Text herausgezogen sein. Durch die in Tabelle 3.2 als *NLP*2 bezeichneten *NLP* kann man spezialisierte Information aus den medizinischen Datenbanken extrahieren. Solche *NLP*2 werden breit, zum Beispiel, für die Bildung der medizinischen Datenbanken verwendet.

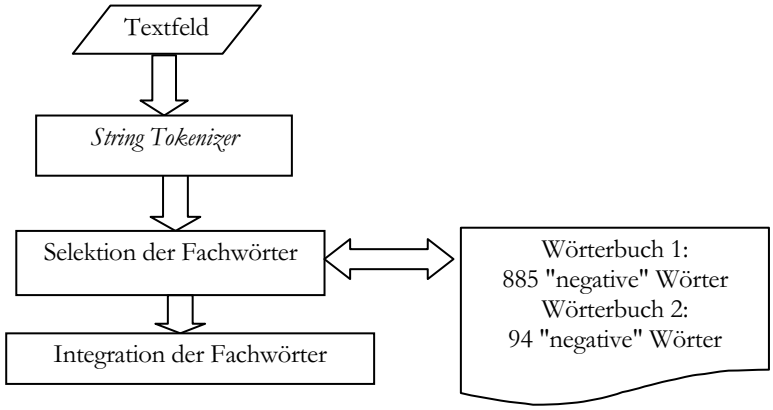

Abbildung 3.7. Fachwörterextraktion

Die Algorithmen *NLP2* suchen bestimmte Wörter und Ausdrücke. Als medizinische Texte werden z.B. die *Abstracts* aus *PubMed* verwendet. Auch die *NLP2* sind für die Regelvalidierung ungeeignet, weil mit der Suche nach bestimmten Wörter die nötige Flexibilität nicht gegeben ist. Außerdem ist die Liste der universellen Ausdrücke ständig zu korrigieren und deren Erzeugung eine schwere Aufgabe.

Es ist folglich Erarbeitung eines anderen Algorithmus - *NLP3* genannt - erforderlich. Ein solcher *NLP* soll die Liste aller Ausdrücke aus dem spezialisierten Text generieren.

Name	Feld	Ziel	Quelle
NLP1	universelle	alle Fachwörter	MetaMap[Aronson2001], GENIA[Kim2003]
NLP2	begrenzte	begrenzte	[deBruijn2002, Albert2003, Chiang2004]
NLP3	begrenzte	alle wichtige Fachwörter	-

Tabelle 3.2. *NLP* (Natural Language Processors), die in der medizinischen und biologischen Forschung verwendet werden

*NLP*3 wurde folgendermaßen konstruiert. Der Algorithmus *NLP*3 ist ähnlich zu *NLP*1. *NLP*3 markiert die Wörter und sucht sie in den Listen der "negativen" Wörter (s. Abb. 3.7). Im Unterschied zu vielen *NLP*1 vereinigt *NLP*3 die benachbart stehenden Wörter, wenn sie sich nicht in der "negativen" Liste befinden. Beispielsweise für den Text aus dem Feld CC von *Uniprot/Swiss-Prot/TrEMBL*

«CC -!- FUNCTION: PROTON-CONDUCTING PORE FORMING SUBUNIT OF THE MEMBRANE INTEGRAL V0 COMPLEX OF VACUOLAR ATPASE»

werden die folgenden Schlüsselwörter generiert:

1. „PROTON-CONDUCTING PORE FORMING“,
2. „MEMBRANE INTEGRAL“,
3. „VACUOLAR ATPASE“.

Die Programmierung des *NLP*3 Algorithmus ist nicht kompliziert, aber die Zusammenstellung der Liste der "negativen" Wörter war schwierig. Für die Zusammenstellung dieser Negativ-Liste wurde im Rahmen der vorliegenden Arbeit ein Programm geschrieben, das die statistische Analyse der Wörter im gesamten *Uniprot/Swiss-Prot/TrEMBL* leistet und dabei einige logische Regeln verwendet. Mit diesen Regeln werden alle Verben, Adjektive und eine große Zahl von Substantiven ausgeschlossen. Falsch gewählte Schlüsselwörter und Wortverbindungen wurden bei der Nutzung sofort sichtbar, so dass eine wiederholte Korrektur der Negativliste bzw. der Schlüsselwort-Listen im Rahmen der vorliegenden Arbeit nötig und möglich war. Im Anhang werden die schließlich erhaltenen Listen aufgeführt.

3.2. Netzwerkanalyse

Es wäre interessant, Netzwerke der Genwechselwirkungen nach dem folgenden Schema zu analysieren:

1. Auswahl der exprimierten Gene;
2. Regelextrahierung;
3. Konstruktion eines Netzwerkes der Genwechselwirkungen aufgrund der bekannten Tatsachen;
4. Regelanalyse mit Hilfe des erhaltenen Netzwerkes.

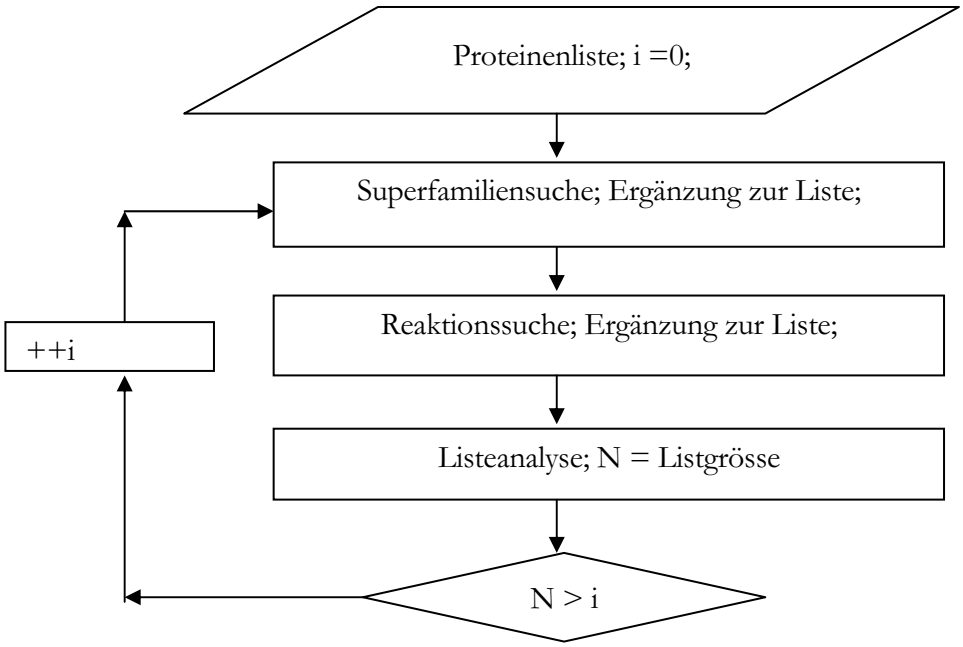

Abbildung 3.8. Schema der Erzeugung von Signalpathways

Für die Realisierung dieser Herangehensweise wurde im Rahmen der vorliegenden Arbeit die Datenbank „*TRANSPATH*“ [Krull2003] genutzt. Das Schema der Erzeugung von *Signalpathways* aufgrund der Interaktionen aus *Transpath* ist in Abbildung 3.8 dargestellt. Der Algorithmus arbeitete effektiv und erlaubt es, Netzwerke aufgrund der in den Regeln als Merkmale vorkommenden in die akzeptable Zeit zu konstruieren. Das Programm nutzte die Liste der Gene und die Zahl *n* (die maximale Entfernung zwischen zwei Genen) als die Inputparameter. Das Programm generierte die Netzwerk der Genwechselwirkungen in der Form der *Pajek*-Inputdatei (http://vlado.fmf.uni-lj.si/pub/networks/pajek).

Bei dieser Herangehensweise entstand folgendes Problem: Beim kleinen *n* Wert stimmten die theoretische Ergebnisse und die erhaltenen experimentellen Ergebnissen nicht immer überein, d.h. die meisten Regeln wurden verworfen. Bei einer zu großen Zahl *n* entsteht eine zu große Zahl möglicher Kombinationen deren Bearbeitung zu kompliziert ist. In Abbildung 3.9 ist das *Signalpathway*-Netzwerk für 10 Proteine und $n = 8$ vorgestellt. Obwohl die erhaltenen Ergebnisse interessant sein können, sind sie offensichtlich für die Durchführung der Regelmengenvalidierung ungeeignet und verlangen die Erarbeitung der Reihe speziellen Methoden. Dies konnte im Rahmen der vorliegenden Arbeit in der vorgegebenen Zeit nicht geleistet werden.

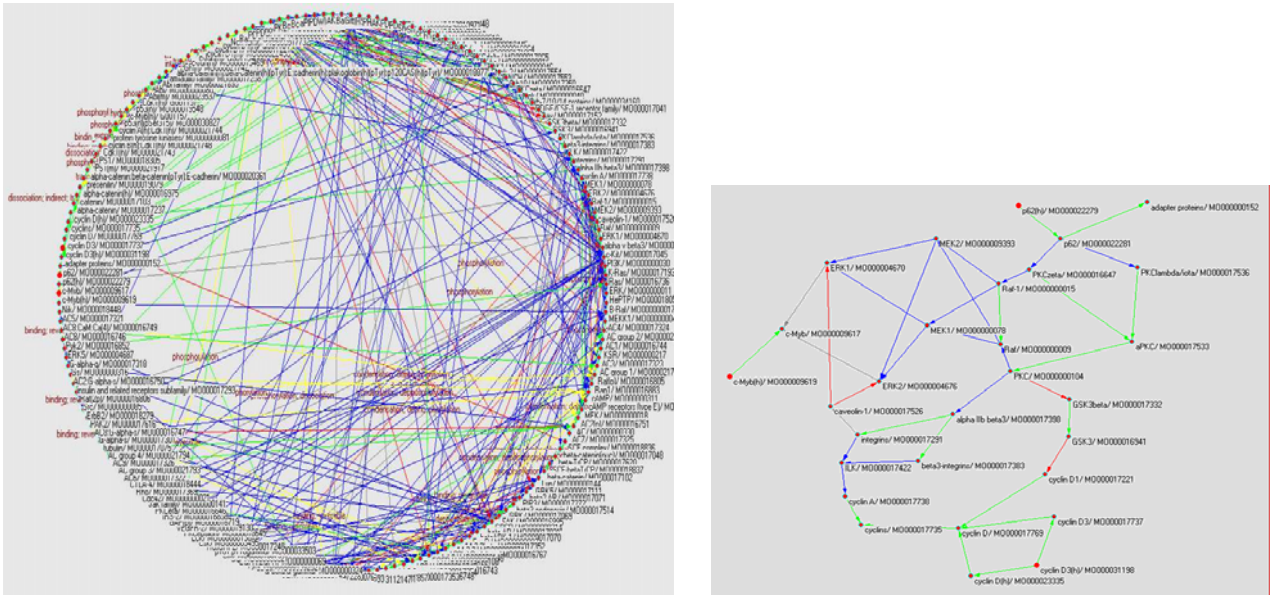

Abbildung 3.9. Ein Beispiel von Signaltransduktionpathways. Links $n = 8$ und rechts für die gleichen Proteine $n = 5$.

3.3. Fazit

1. Die in der Literatur beschriebenen Methoden der Textanalyse sind für die Aufgabe der Regelvalidierung ungeeignet. Es war die Entwicklung einer neuen Methode erforderlich.

2. Die neue Regelvalidierungsmethodik basiert auf Schlüsselwörtern (Beschreibungen), die in Datenbanken den Merkmalen (Genen) zugeordnet sind. Diese Methodik hat es ermöglicht, die Anzahl der Merkmale in Regeln zu verringern. Für die Realisierung der Methodik war die Erarbeitung eines speziellen Typs von *NLP* erforderlich.

3. Die Anwendung der neuen Methode ermöglicht – wie noch an drei Beispielen gezeigt werden wird :
   - ein besseres Verstehen der Regeln
   - den Gewinn von neuer Information
   - die Verringerung der Anzahl und die Verbesserung der Qualität der Regeln.

Für die Analyse der Regeln aufgrund von Protein-Protein-Wechselwirkungen, wie sie in der Datenbank *Transpath* abgelegt sind, wurde eine neue Methodik entwickelt. Aufgrund der Komplexität der automatischen Analyse konnten jedoch keine brauchbaren Ergebnisse erhalten werden.

4 Materialien, Technologien und Datenbearbeitungsmethode

4.1.Materialien und Technologien

4.1.1. Software und Technologien

- Für die Implementierung der Algorithmen zur Regelextraktion wurde die Programmiersprache *Java* 2 (*Java* 1.4), sowie *JBuilder* bis zur Version X (Personal/Foundation) verwendet[Weber2000][Horstmann2003a][Horstmann2003b]..
- *XML (JDOM)* wurde als Sprache für die Interaktion der verschiedenen Programmteile benutzt[Pitts2000].
- *Oracle 9* in Verbindung mit *SQL* wurde wie als lokale Datenbank benutzt [Page2000].
- *Pajek* wurde für die graphische Darstellung der Ergebnisse benutzt. Ein ini-Datei-Generator für *Pajek* ist entwickelt worden.

4.1.2. Biologische Datenbasen

- *Uniprot/Swiss-Prot/TrEMBL*. Informationen über Proteine (www.expasy.ch)
- *Transpath* [Krull2003]. Signaltransduktionsreaktionen (www.biobase.de)
- *KEGG* [Kanehisa2002]. Metabolische Reaktionen und Lokalisierungen in den Zellprozessen (www.kegg.com).

4.1.3. Die Datensätze

In der bioinformatischen Literatur verwenden die Autoren häufig nur eine einzige Methodik oder/und einen einzigen Datensatz. Diese Methoden sind für Klassifikationsaufgaben traditionell *k-NN*, lineare Diskriminanzanalyse und Entscheidungsbäume. Es ist nicht möglich, eine Methodik ohne Vergleichsanalyse mit verschiedenen konkurrierenden Methoden anhand mehrerer Datensätzen zu bewerten. [Dudoit2000].

In der vorliegenden Arbeit wird nachfolgend die neu entwickelte Methode anhand von drei verschiedenen Datensätze getestet und die erhaltenen Ergebnisse werden mit den Ergebnissen anderer Autoren verglichen.

Der erste Datensatz „Leukämie“ wird genutzt, weil er am häufigsten zitiert wird. Die anderen zwei Datensätze werden nachfolgend genutzt, weil sie die schlechtesten Ergebnisse bei der Anwendung vorhandener Mustererkennungsalgorithmen zeigten.

Zur Vergleichbarkeit der verschiedenen Ergebnisse werden bei allen drei Datensätzen dieselben Inputparameter verwendet. Es sei aber bemerkt, dass für den zweiten und dritten Datensatz die Nutzung anderer, angepasster Parameter die Ergebnisse bedeutend verbessern kann.

4.1.3.1. Datensatz „Leukämie“

Akute Leukämie ist eine Krankheit von den Leukozyten und ihren Vorläufern. Bei der Leukämie gibt es unreife, abnormale Zellen im Knochenmark, peripherem Blut und häufig in der Leber, Milz, Lymphknoten und anderen parenchymatischen Organen. Akute Leukämie ist laut der morphologischen, zytochemischen und immunologischen Kriterium klassifiziert (http://www.meds.com/leukemia). Obwohl einige genetische Defekte mit *AML* verbunden sind, ist ein großer Prozentsatz der *AML* - Fälle zytogenetisch „normal“ [Reuther2002]. Die zugrunde liegenden genetischen Defekte der *AML*-Pathogenese sind immer noch unbekannt [Erkeland2003].

Der Datensatz „Leukämie“ besteht aus Sätzen für das Lernen und für die Testung, um *ALL* und *AML* auf der Basis von den Expressionen von 6817 Genen vorherzusagen [Golub1999].

Die originalen Lerndaten bestehen aus 27 *ALL*- und 11 *AML* - Fällen von 38 Knochenmarkproben (von erwachsenen Patienten). Der Kontrolldatensatz besteht aus 24 Knochenmarkproben sowie 10 peripheren Blutproben von Erwachsenen und Kindern (20 *ALL*- und 14 *AML* - Proben).

Die genaue Untereinteilung der Leukämie und die Identifikation von prognostischen Markern sind wesentlich für die verbesserte Diagnostik und Therapie. Die Analyse dieser Daten soll eine mögliche künftige Anwendung der Arraytechnologie bei der Klassifizierung der akuten Leukämie unterstützen [Kohlmann2003].

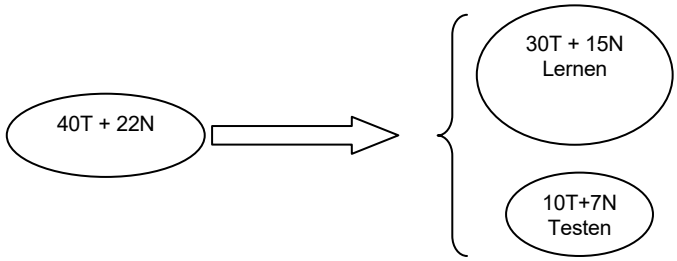

Abbildung 4.1. Die Verteilung der Darmkrebs Daten

4.1.3.2. Datensatz „Darmkrebs“

Der Datensatz besteht aus 40 Tumor- und 22 Normal - Kolongeweben (2000 Gene). Die Autoren [Alon1999] verwendeten einen gemeinsamen Datensatz für das Lernen und die Testung der Klassifikationsgüte mittel *Cross-Validierung*. Im Rahmen der vorliegenden Arbeit wurde zur Einschätzung der Klassifikationsgüte der Datensatz aufgeteilt in den Lern- und den Testsätze, wobei die ersten zwei Drittel der Objekte für das Lernen und das dritte Drittel für das Testen verwendet werden. Das Schema der Datenteilung ist in Abbildung 4.1. dargestellt.

4.1.3.3. Datensatz „Hirntumor“

Es wurde für die nachfolgende Analyse der Datensatz C von [Pomeroy2002] benutzt mit 60 Proben, davon 39 Medulloblastom - Überlebende und 21 von Behandlungsmisserfolgen.

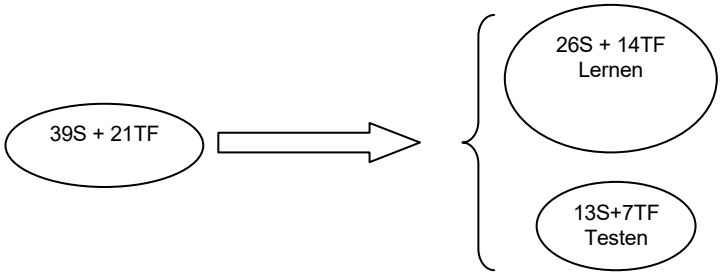

Abbildung 4.2. Die Verteilung der Hirntumor Daten

Für jede Untersuchung liegen Expressionswerte von 7129 Genen vor. Autoren benutzten für die *Cross-Validierung* den gesamten Daten-Satz (60 Proben für das Lernen und 1 Probe für das Testen nach der „*leave-one-out*“ Methode). In der vorliegenden Arbeit werden die Daten in einen Lern- und einen Testsatze aufgeteilt: wieder die erste zwei Drittel der Proben für das Lernen und das verbleibende Drittel für das Testen. Die Teilung des Datensatzes ist in Abb. 4.2 dargestellt.

4.2. Bearbeitung der biologischen Datenbanken

In der Arbeit werden folgende 3 Datenbanken verwendet: *UniProt/Swiss-Prot/TrEMBL*, *Transpath* und *KEGG*. Alle drei Datenbanken sind in Internet zugänglich. Auf den entsprechenden Seiten existieren komfortable Möglichkeiten der Suche nach Proteinen oder Genen. Die Suchergebnisse können in verschiedenen Formaten (*HTML, XML, SCV* usw.) ausgegeben werden. Das Dateiformat ist gut protokolliert, so dass die Bearbeitung der Suchergebnisse nicht schwierig ist.

Es gibt 2 Möglichkeiten der Datenbearbeitung: ein lokaler Zugriff und ein Zugriff durch das Internet. Die Ausarbeitung des Programms für den Datenbankzugriff über das Internet mit dem Protokoll *TCP/IP* ist auf der Basis der Programmiersprache *Java* 2 nicht schwierig. Die Bearbeitung der Daten, die in die lokale *SQL* - Datenbank importiert werden, erfolgt in zwei Etappen:

- Die Textdatei wird vom Server geladen. Danach wird sie bearbeitet und in die lokale *SQL* Datenbank importiert.
- *SQL* Datenbank wird benutzt.

Bei der Arbeit mit Datenbanken sind folgende Probleme zu beachten:

- Die Zugriffszeit zum *WWW* Server ist im Vergleich zum Zugriff auf die lokalen *SQL* Datenbank durch *JDBC/ODBC* nicht genügend hoch. Wenn die Anzahl der Abfragen nicht hoch ist, spielt die Zugriffszeit keine Rolle. Im Fall von Dutzenden (für *UniProt/Swiss-Prot/TrEMBL*) und Hunderte oder Tausende (für *KEGG* und *Transpath*) Abfragen ist es nicht möglich den Zeitparameter zu ignorieren.
- In der Arbeit wird nur ein Teil der Information, die mit der *HTML*-Datei vom Server geladen wird, verwendet. Die Hauptarbeit ist die Extraktion der notwendigen und die Entfernung der nutzlosen Information.
- Die Nutzung der lokalen *SQL* Datenbank kann die informative Analyse bedeutend verbessern.

Deshalb wird die, zweite Variante der Datenbearbeitung benutzt, das heißt die lokale *SQL* Datenbank (s. Abb. 4.3). Die Struktur der Datenbanktabellen ist unten in den Abbildungen 4.4-4-6 dargestellt.

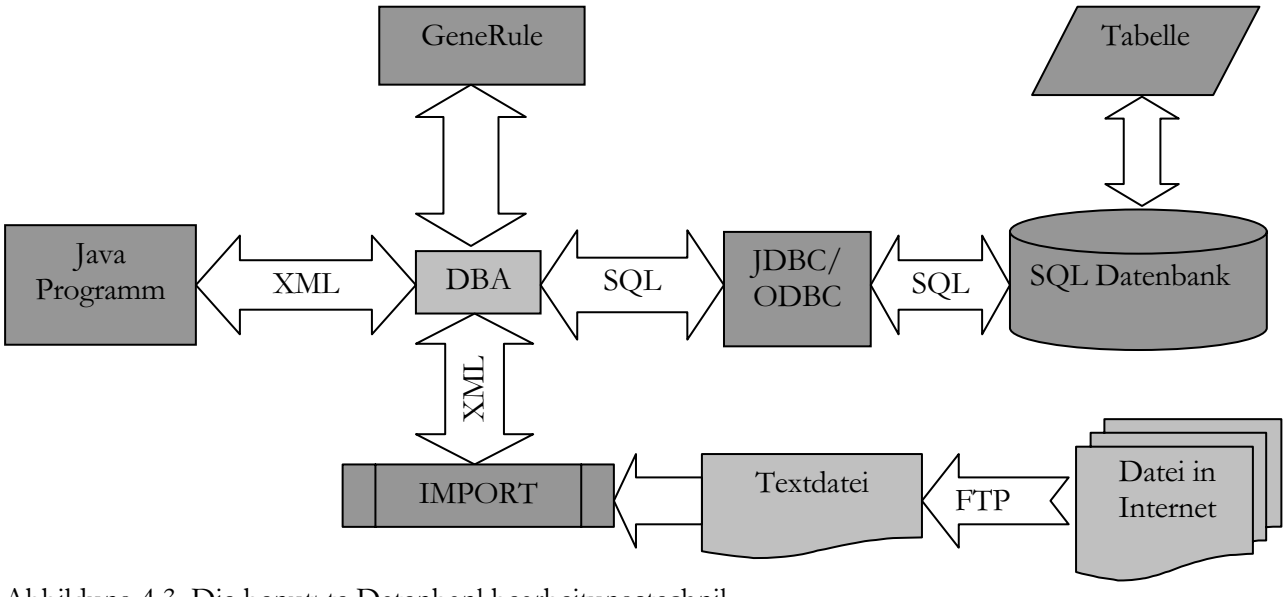

Abbildung 4.3. Die benutzte Datenbankbearbeitungstechnik.

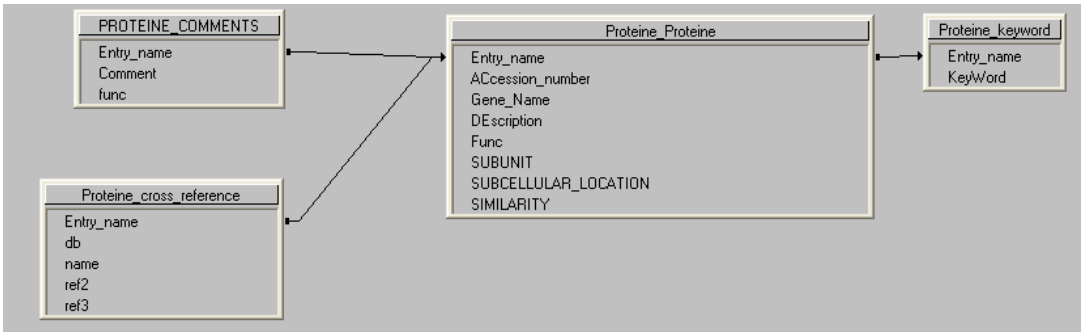

Abbildung 4.4 *UniProt/Swiss-Prot/TrEMBL* Tabellen in den lokalen Datenbanken

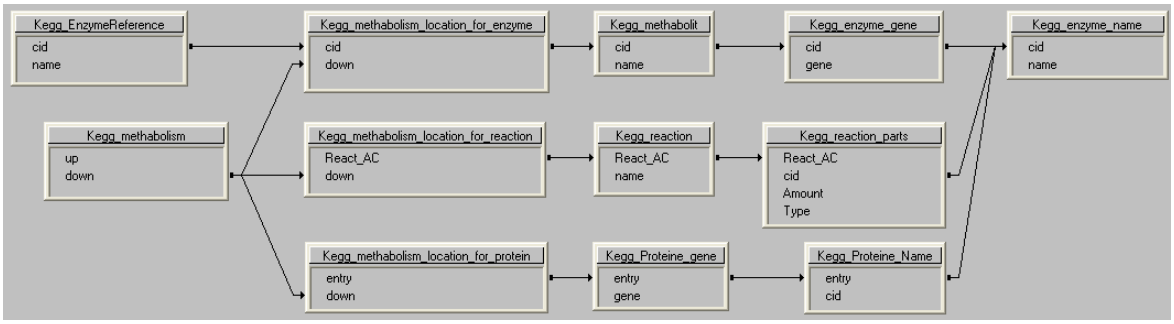

Abbildung 4.5 *KEGG* in den lokalen Datenbanken

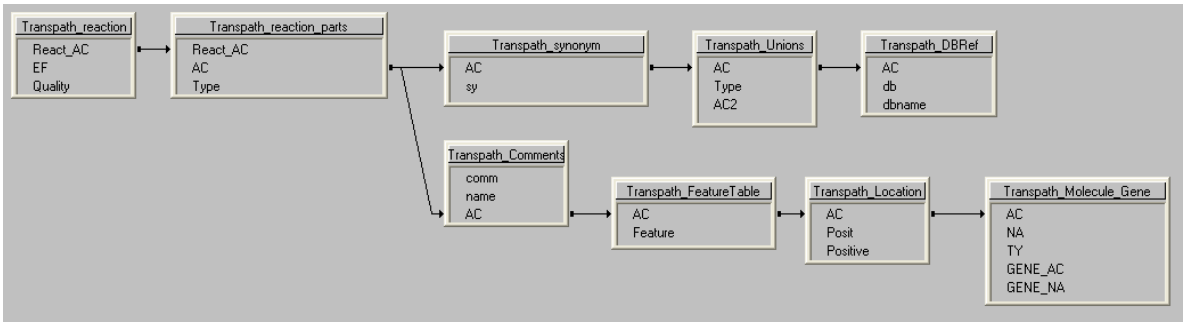

Abbildung 4.6 *Transpath* in den lokalen Datenbanken

4.3. Merkmale von *GeneRule* Version 2.0

Die Hauptmerkmale von *GeneRule* 2.0 sind:

- Datendiskretisierung: Sigma, mit/ohne Logarithmierung/Minimum
- Datenselektion: Prozent von Werten
- Analysis von selektierten Variablen
- Parameter der Regelgenerierung
  - min. Anzahl von Patienten in einer Klasse
  - max. Anzahl von falsch klassifizierten Patienten
- Analysis und Nachbearbeitung von Regeln
- *DNF*- Erzeugung
- Regelbasierte Clusterbildung
- Analysis der Variablen (*UniProt/Swiss-Prot/TrEMBL* und *Transpath*)
- Zugriff zu *UniProt/Swiss-Prot/TrEMBL*, *Transpath* und *KEGG* Datenbanken
- Anwendung auf Lern- und Test-Daten und Gütebewertung der Regeln bzw. Daten

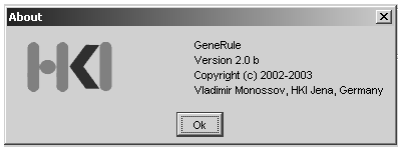

Abbildung 4.7 Info - Panel von GeneRule

In der Abbildung 4.8 ist das allgemeine Arbeitsschema von *GeneRule* dargestellt.

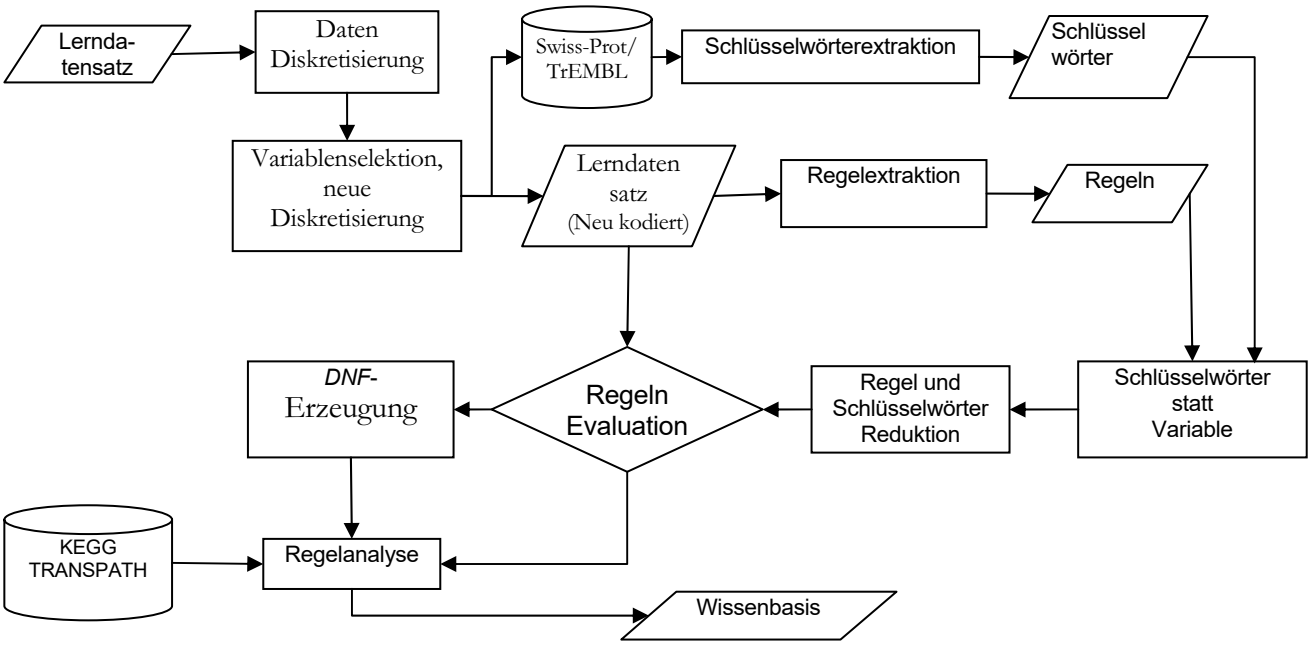

Abbildung 4.8. Schema der Datenbearbeitung.

Screenshots von *GeneRule* (Abb. 4.9 -4.12).

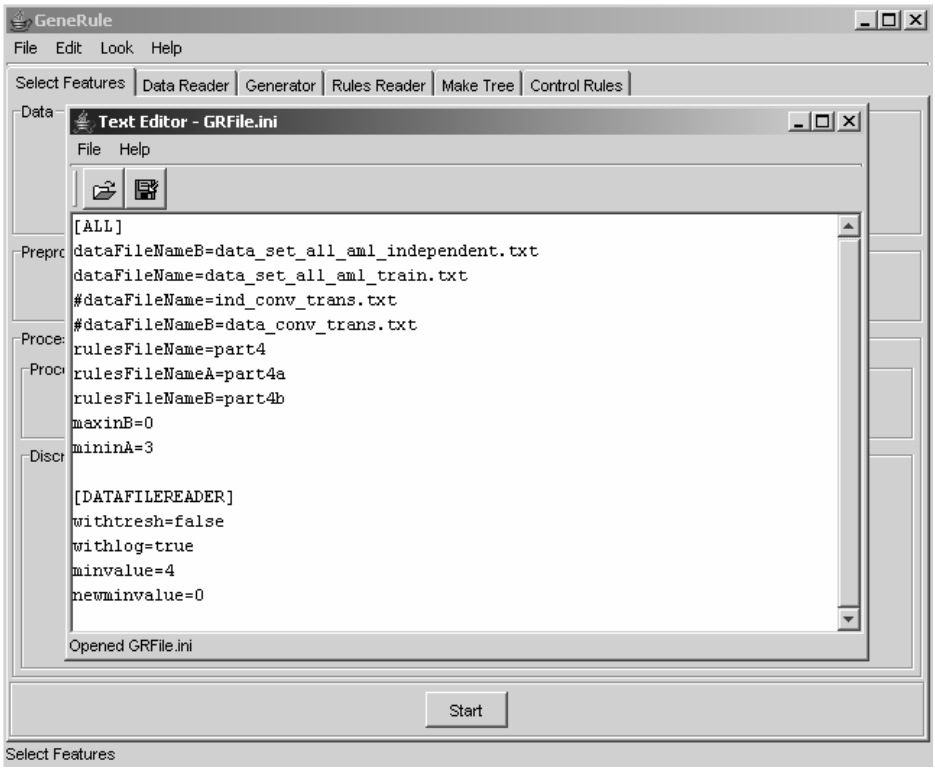

Abbildung 4.9. *GeneRule* Panel (vorn ist eine ini-Datei).

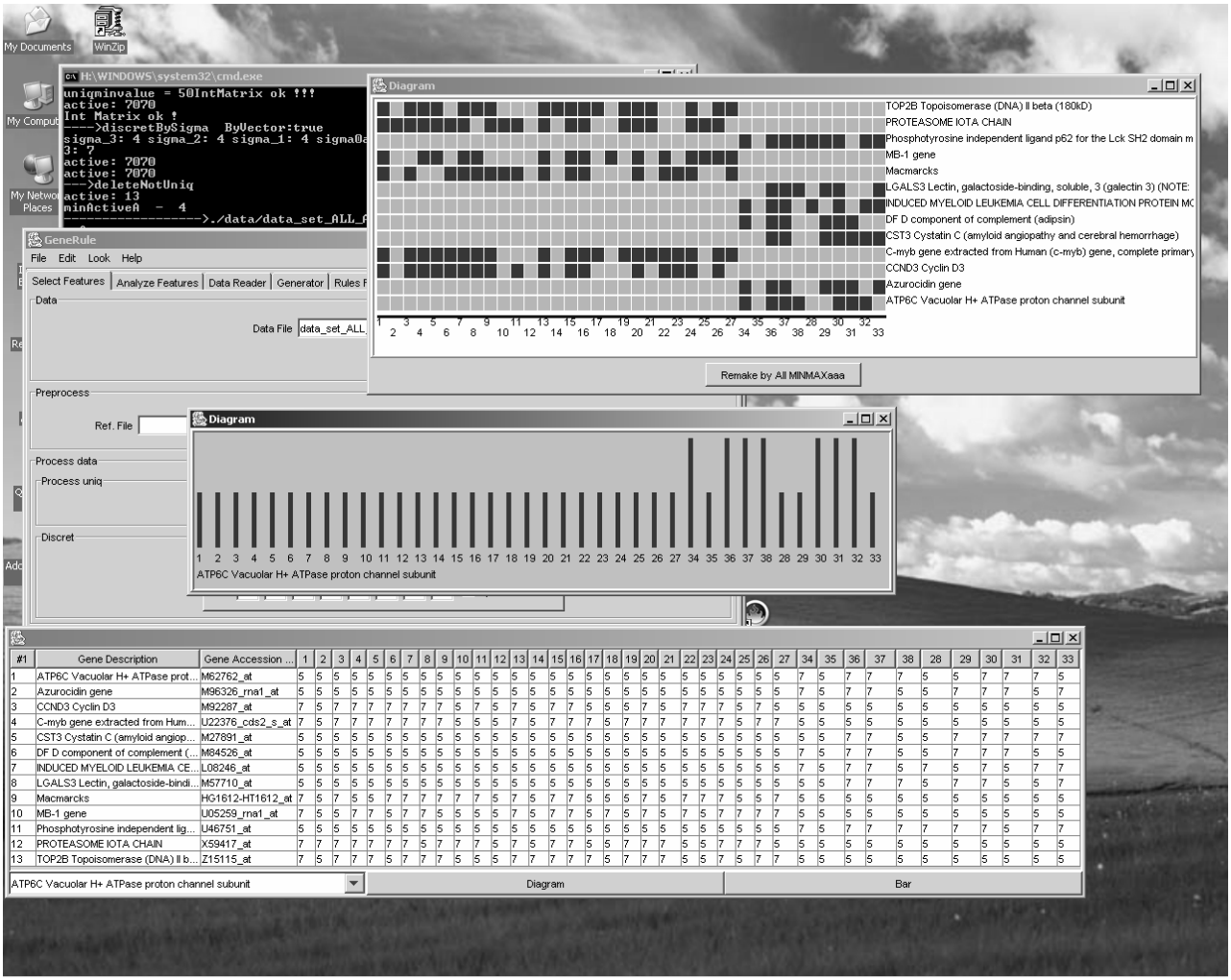

Abbildung 4.10. Panelen nach der Selektion von Merkmalen

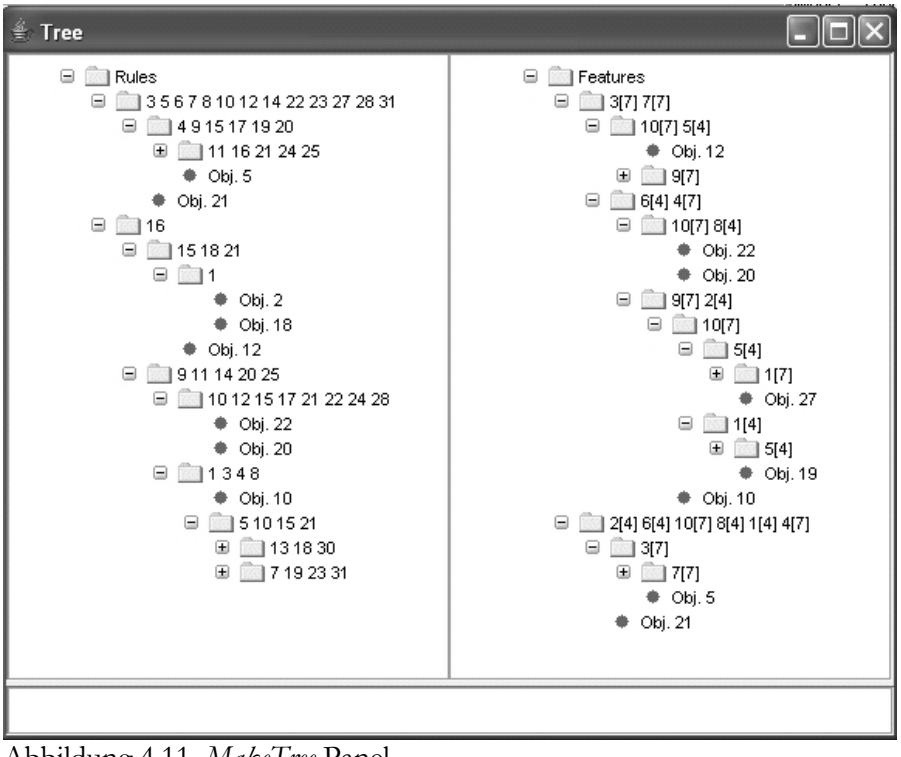

Abbildung 4.11. *MakeTree* Panel

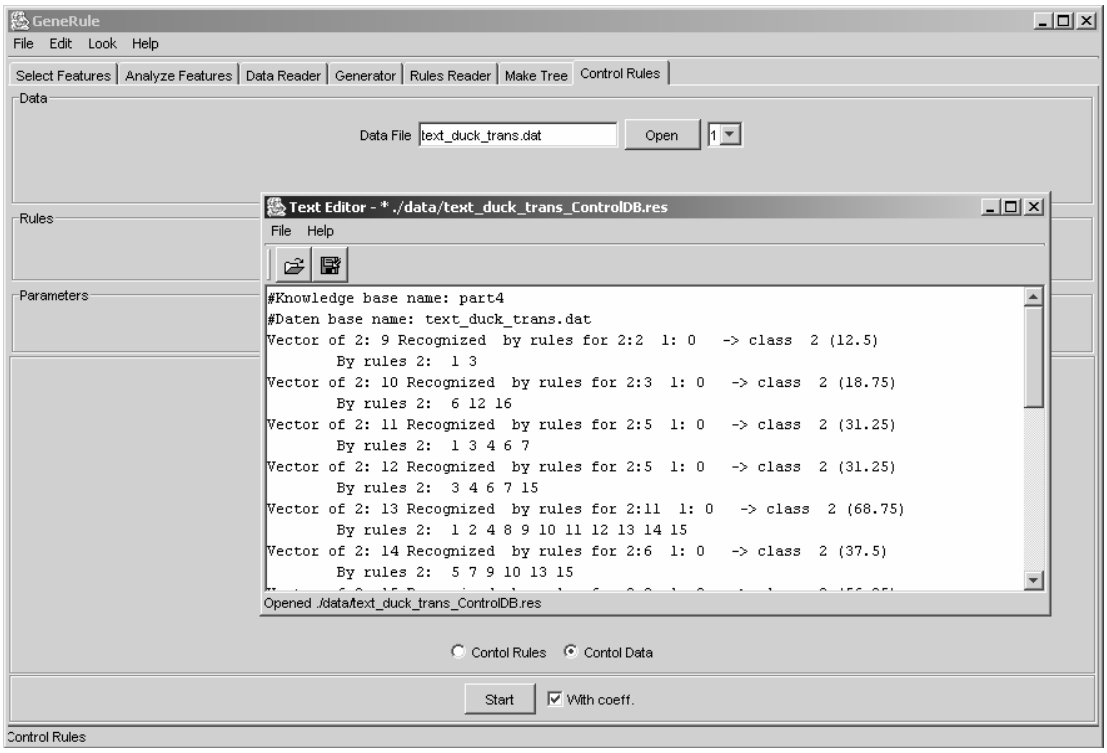

Abbildung 4.12. *ControlRules* Panel nach der Analyse

5 Ergebnisse

5.1. Datensatz „Leukämie“

5.1.1. Auswahl der Gene

In der ersten Etappe wurden die Merkmale gewählt, die den verschiedenen Schwellen Sc entsprechen (Abb. 5.1). Die neuen logischen Regeln wurden hinsichtlich der Erkennungsqualität (Anzahl der richtigen Erkennungen Λ im Trainingsatz) verglichen. Die Ergebnisse befinden sich in der Tabelle 5.1. Die Erkennungsqualität der logischen Regeln hängt von der Merkmalanzahl bei Schwelle 60 % und niedriger nicht ab, deshalb kann man die Schwelle 60 % verwenden. Die Anzahl der Merkmale ist dann 10.

Sc	Merkmalanzahl	Regelanzahl (*ALL*/*AML*)	Anzahl der richtigen Erkennungen Λ
30%	66	974 (562/412)	38 von 38
40%	29	236 (126/110)	38 von 38
50%	13	60 (31/29)	38 von 38
60%	10	51 (31/20)	38 von 38
70%	3	-	-

Tabelle 5.1. Die Erkennungsqualität je nach der Schwelle Sc.

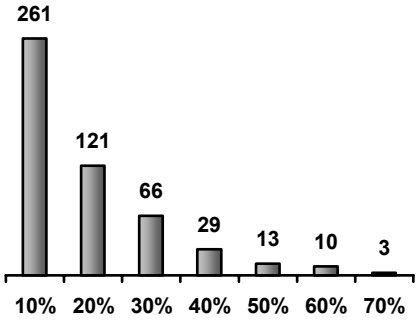

Abbildung 5.1. Die Merkmalanzahl für verschiedene Schwellen Sc

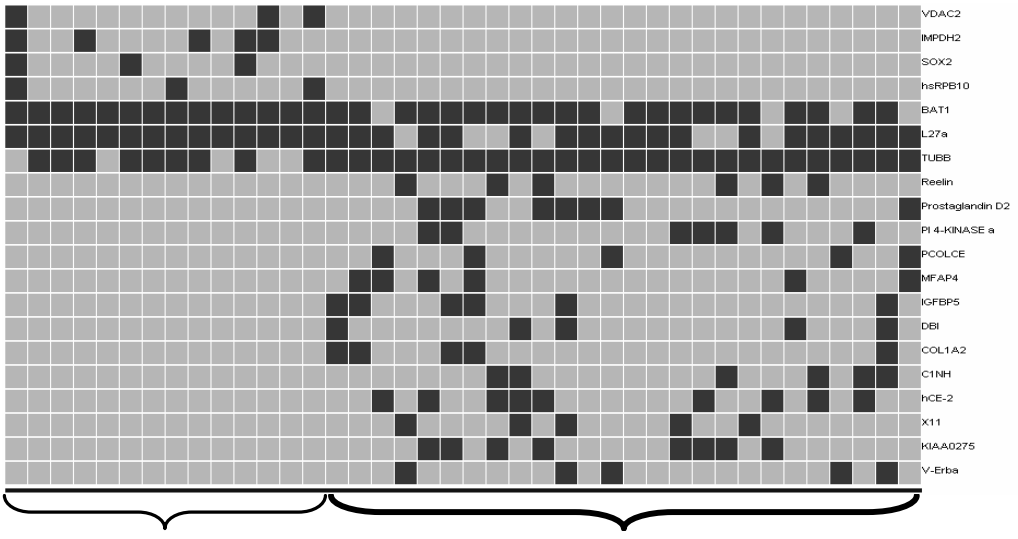

Abbildung 5.2. Die graphische Darstellung der ausgewählten Gene. Die hochexprimierte Gene sind schwarz. Rechts ALL, links AML.

Die ausgewählten Gene befinden sich in der Tabelle 5.2 und die graphische Darstellung der gewählten Gene ist in Abbildung 5.2 gezeigt. Man kann eine deutliche Grenze zwischen den hochexprimierten Genen bei *ALL* und *AML* sehen. Die ausgewählten Gene wurden auch von anderen Autoren ausgewählt, wie die Tabelle 5.3 zeigt

*	Gennamen	Swiss-Prot/TrEMBL Namen
1	ATP6C	VATL_HUMAN
2	Azurocidin	CAP7_HUMAN
3	Cyclin D3	CGD3_HUMAN
4	C-myb	MYB_HUMAN
5	Cystatin C	CYTC_HUMAN
6	MCL1	MCL1_HUMAN
7	Macmarcks	MRP_HUMAN
8	p62	Q13501
9	Das Jota	PSA6_HUMAN
10	TOP2B	TP2B_HUMAN

Tabelle 5.2. Ausgewählte Gene

In der Literatur kann man die folgende Information über die Leukämie-relevanten Gene finden: [Chow2001]: *«Cystatin C, azurocidin, and adipsin, are good targets for investigating the biology of ALL/AML».* [Calabretta1991]: *«The c-myb proto-oncogene is preferentially expressed in hematopoietic cells, and its encoded protein, Myb, is required for hematopoietic cell proliferation» [*Bloch1995]: *«C-myb and c-ets-1 have to function as proto-oncogenes. Stimulated expression of such transcription factor can then lead to the continuous proliferation cycle characteristic of the cancer cell».* [Mavilio1986]*: «Studies of c-myb, c-myc, c-fos and c-fms showed no gross genetic alterations, amplifications or variations in mRNA transcripts deriving from these genes. Expression of c-myc and c-myb was detected in all leukemic cells at variable levels and was characterized by well-defined patterns within all subtypes. Cellular oncogene expression in specific subtypes of leukemic cells may relate the proliferative activity (c-myc, c-myb) ».*

Die Gene	[Golub99]	[Thomas01]	[Bijani03]
ATP6C	+	+	
Azurocidin	+		+
Cyclin D3	+	+	
C-myb	+	+	
Cystatin C	+		+
Macmarcks		+	
p62	+		
Iota	+	+	
MCL1	+		
TOP2B	+	+	
Die Summe	9	6	2

Tabelle 5.3. Ausgewählte Gene der anderen Autoren

Nach der nochmaligen Kodierung (nur aufgrund der gewählten Gene) ist das Bild nicht so deutlich, wie auf der Abbildung 5.2. Das Problem kann durch die Nutzung der logischen Regeln gelöst werden.

5.1.2. Die Regelextraktion

In Abb. 5.3 sind die extrahierten Regeln dargestellt. Für den Datensatz "Leukämie" wurden 51 Regeln gefunden. Mit dem Regelsatz können alle Objekte vom Lerndatensatz korrekt erkannt werden. Die Anwendung auf den Kontrolldatensatz ergibt eine falsche Vorhersage.

Im nachfolgenden Abschnitt werden die Möglichkeiten der Wissensextraktion und der weiteren Bearbeitung der logischen Regeln untersucht. Die logischen Regeln sind durch die *DNF*-Erzeugung extrahiert worden. Deshalb ist die weitere Vereinfachung der Regeln nicht möglich. Auch durch die Nutzung von Expertenwissen konnte keine Vereinfachung erzielt werden. Der Versuch, das Problem durch die Nutzung der automatischen Clusterbildung zu lösen, ergab keine befriedigenden Ergebnisse; da keine stabilen Cluster gefunden werden konnten.

In Abb. 5.4 sind die Programmergebnisse, die erzeugten logischen Regeln; von *DataControl* dargestellt. Das Erkennensbild ist ziemlich kompliziert. Wahrscheinlich wird die Nutzung von verschiedenen Methoden der Clusterbildung kein eindeutig genaues Ergebnis erzielen. Es gibt verschiedene Einteilungsmöglichkeiten der Objekte (siehe Clusterergebnissen für Datensatz "Gesichter"). Deshalb wird die Anwendung von Methoden der Clusterbildung keine befriedigenden Ergebnisse für diesen Datensatz liefern. Im folgenden Abschnitt wird die Anwendbarkeit der logischen Methode untersucht.

#Knowledge base name: part4
#Data base name: data_set_all_aml_train.dat
#In Class ALL : 27, in another Class : 11
-->>1r 9[h] in ALL: 24(88%) in non ALL: 0(0%)
-->>2r 1[h] 6[l] in ALL: 3(11%) in non ALL: 0(0%)
-->>3r 2[l] 6[l] in ALL: 23(85%) in non ALL: 0(0%)
-->>4r 2[l] 3[h] in ALL: 23(85%) in non ALL: 0(0%)
-->>5r 2[l] 10[h] in ALL: 23(85%) in non ALL: 0(0%)
-->>6r 2[l] 8[l] in ALL: 17(62%) in non ALL: 0(0%)
-->>7r 2[l] 1[l] in ALL: 19(70%) in non ALL: 0(0%)
-->>8r 2[l] 4[h] in ALL: 23(85%) in non ALL: 0(0%)
-->>9r 6[l] 3[h] in ALL: 23(85%) in non ALL: 0(0%)
-->>10r 6[l] 10[h] in ALL: 23(85%) in non ALL: 0(0%)
-->>11r 6[l] 7[h] in ALL: 22(81%) in non ALL: 0(0%)
-->>12r 6[l] 8[l] in ALL: 17(62%) in non ALL: 0(0%)
-->>13r 6[l] 5[l] in ALL: 13(48%) in non ALL: 0(0%)
-->>14r 6[l] 4[h] in ALL: 24(88%) in non ALL: 0(0%)
-->>15r 3[h] 10[h] in ALL: 25(92%) in non ALL : 0(0%)
-->>16r 3[h] 7[h] in ALL: 25(92%) in non ALL: 0(0%)
-->>17r 3[h] 8[l] in ALL: 17(62%) in non ALL: 0(0%)
-->>18r 3[h] 5[l] in ALL: 16(59%) in non ALL: 0(0%)
-->>19r 3[h] 1[l] in ALL: 20(74%) in non ALL: 0(0%)
-->>20r 3[h] 4[h] in ALL: 24(88%) in non ALL: 0(0%)
-->>21r 10[h] 7[h] in ALL: 24(88%) in non ALL: 0(0%)
-->>22r 10[h] 8[l] in ALL: 18(66%) in non ALL: 0(0%)
-->>23r 10[h] 1[l] in ALL: 20(74%) in non ALL: 0(0%)
-->>24r 7[h] 8[l] in ALL: 16(59%) in non ALL: 0(0%)
-->>25r 7[h] 4[h] in ALL: 23(85%) in non ALL: 0(0%)
-->>26r 8[l] 5[l] in ALL: 8(29%) in non ALL: 0(0%)
-->>27r 8[l] 1[l] in ALL: 16(59%) in non ALL: 0(0%)
-->>28r 8[l] 4[h] in ALL: 17(62%) in non ALL: 0(0%)
-->>29r 5[l] 1[l] in ALL: 11(40%) in non ALL: 0(0%)
-->>30r 5[l] 4[h] in ALL: 14(51%) in non ALL: 0(0%)
-->>31r 1[l] 4[h] in ALL: 21(77%) in non ALL: 0(0%)

#Knowledge base name: part4
#Data base name: data_set_all_aml_train.dat
#In Class AML : 11, in another Class : 27
-->>1r 5[h] in AML: 9(81%) in non AML: 0(0%)
-->>2r 9[l] 10[l] in AML: 9(81%) in non AML: 0(0%)
-->>3r 9[l] 3[l] in AML: 10(90%) in non AML: 0(0%)
-->>4r 9[l] 7[l] in AML: 8(72%) in non AML: 0(0%)
-->>5r 1[h] 10[l] in AML: 7(63%) in non AML: 0(0%)
-->>6r 1[h] 3[l] in AML: 8(72%) in non AML: 0(0%)
-->>7r 1[h] 7[l] in AML: 7(63%) in non AML: 0(0%)
-->>8r 10[l] 2[h] in AML: 7(63%) in non AML: 0(0%)
-->>9r 10[l] 3[l] in AML: 8(72%) in non AML: 0(0%)
-->>10r 10[l] 6[h] in AML: 8(72%) in non AML: 0(0%)
-->>11r 10[l] 7[l] in AML: 6(54%) in non AML: 0(0%)
-->>12r 10[l] 4[l] in AML: 9(81%) in non AML: 0(0%)
-->>13r 2[h] 3[l] in AML: 8(72%) in non AML: 0(0%)
-->>14r 2[h] 7[l] in AML: 8(72%) in non AML: 0(0%)
-->>15r 3[l] 6[h] in AML: 9(81%) in non AML: 0(0%)
-->>16r 3[l] 4[l] in AML: 9(81%) in non AML: 0(0%)
-->>17r 3[l] 8[h] in AML: 9(81%) in non AML: 0(0%)
-->>18r 6[h] 7[l] in AML: 7(63%) in non AML: 0(0%)
-->>19r 7[l] 4[l] in AML: 7(63%) in non AML: 0(0%)
-->>20r 7[l] 8[h] in AML: 7(63%) in non AML: 0(0%)

Abbildung 5.3. Die erzeugten logischen Regeln (*GeneRule* log-Datei). Es sind die Regeln für *ALL* rechts und für *AML* links dargestellt. Der Bericht fängt mit der Serviceinformation (die Zeilen mit "#" am Anfang) an. Jede der Regeln fängt mit„→“ an, dann folgt die Regelnummer mit dem Endung »r ("Regel") folgt. In der Regel ist jedes Merkmal mittels einer Nummer gekennzeichnet gefolgt von dem Merkmalswert in den quadratischen Klammern. Am Ende steht Λ. Zum Beispiel „->> 1r 9 [h] bei ALL: 24 (88 %) in nicht ALL: 0 (0 %)“: die Regel 1 von der Klasse *ALL* erkennt 88 % *ALL* Fälle und keine Fälle anderer Klasse (die Statistik für die *data_set_all_aml_train.dat*). Die Regel: « Wenn hat das Kennzeichen 9 die Bedeutung „h “, so wird das Objekt auf die Klasse *ALL* bezogen».

```
#Knowledge base name: part4
#Daten base name: data_set_all_aml_train.dat
Vector of AML: 34 Recognized by rules for AML: 12 ALL: 0 -> class AML   By rules AML: 3 4 6 7 13 14 15 16 17 18 19 20
Vector of AML: 35 Recognized by rules for AML:11 ALL: 0 -> class AML   By rules AML: 1 3 4 6 7 13 14 15 17 18 20
Vector of AML: 36 Recognized by rules for AML:20 ALL: 0 -> class AML   By rules AML: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Vector of AML: 37 Recognized by rules for AML:20 ALL: 0 -> class AML   By rules AML: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Vector of AML: 38 Recognized by rules for AML:13 ALL: 0 -> class AML   By rules AML: 1 2 4 5 7 8 10 11 12 14 18 19 20
Vector of AML: 28 Recognized by rules for AML:10 ALL: 0 -> class AML   By rules AML: 2 3 5 6 9 10 12 15 16 17
Vector of AML: 29 Recognized by rules for AML:14 ALL: 0 -> class AML   By rules AML: 1 2 3 4 8 9 11 12 13 14 16 17 19 20
Vector of AML: 30 Recognized by rules for AML:20 ALL: 0 -> class AML   By rules AML: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Vector of AML: 31 Recognized by rules for AML:18 ALL: 0 -> class AML   By rules AML: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 18 19
Vector of AML: 32 Recognized by rules for AML:11 ALL: 0 -> class AML   By rules AML: 1 2 3 5 6 9 10 12 15 16 17
Vector of AML: 33 Recognized by rules for AML:11 ALL: 0 -> class AML   By rules AML: 1 2 3 8 9 10 12 13 15 16 17
 Recognized true 11, error 0, unknown 0 of 11 vectors of AML
Vector of ALL: 1 Recognized by rules for ALL: 30 AML: 0 -> class ALL   By rules ALL: 1 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 - 31
Vector of ALL: 2 Recognized by rules for ALL:12 AML: 0 -> class ALL   By rules ALL: 1 4 5 6 15 16 17 18 21 22 24 26
Vector of ALL: 3 Recognized by rules for ALL:30 AML: 0 -> class ALL   By rules ALL: 1 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 - 31
Vector of ALL: 4 Recognized by rules for ALL:25 AML: 0 -> class ALL   By rules ALL: 1 3 4 5 6 7 8 9 10 11 12 14 15 16 17 19 20 21 22 23 24 25 27 28 31
Vector of ALL: 5 Recognized by rules for ALL:20 AML: 0 -> class ALL   By rules ALL: 1 3 4 5 6 7 8 9 10 12 14 15 17 19 20 22 23 27 28 31
Vector of ALL: 6 Recognized by rules for ALL:30 AML: 0 -> class ALL   By rules ALL: 1 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 -31
Vector of ALL: 7 Recognized by rules for ALL:25 AML: 0 -> class ALL   By rules ALL: 1 3 4 5 6 7 8 9 10 11 12 14 15 16 17 19 20 21 22 23 24 25 27 28 31
Vector of ALL: 8 Recognized by rules for ALL:24 AML: 0 -> class ALL   By rules ALL: 3 4 5 6 7 8 9 10 11 12 14 15 16 17 19 20 21 22 23 24 25 27 28 31
Vector of ALL: 9 Recognized by rules for ALL:30 AML: 0 -> class ALL   By rules ALL: 1 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 -31
Vector of ALL: 10 Recognized by rules for ALL:17 AML: 0 -> class ALL   By rules ALL: 1 3 4 7 8 9 11 13 14 16 18 19 20 25 29 30 31
Vector of ALL: 11 Recognized by rules for ALL:25 AML: 0 -> class ALL   By rules ALL: 1 3 4 5 6 7 8 9 10 11 12 14 15 16 17 19 20 21 22 23 24 25 27 28 31
Vector of ALL: 12 Recognized by rules for ALL:4 AML: 0 -> class ALL   By rules ALL: 15 16 18 21
Vector of ALL: 13 Recognized by rules for ALL:25 AML: 0 -> class ALL   By rules ALL: 1 3 4 5 6 7 8 9 10 11 12 14 15 16 17 19 20 21 22 23 24 25 27 28 31
Vector of ALL: 14 Recognized by rules for ALL:22 AML: 0 -> class ALL   By rules ALL: 1 3 4 5 7 8 9 10 11 13 14 15 16 18 19 20 21 23 25 29 30 31
Vector of ALL: 15 Recognized by rules for ALL:22 AML: 0 -> class ALL   By rules ALL: 1 3 4 5 7 8 9 10 11 13 14 15 16 18 19 20 21 23 25 29 30 31
Vector of ALL: 16 Recognized by rules for ALL:30 AML: 0 -> class ALL   By rules ALL: 1 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24- 31
Vector of ALL: 17 Recognized by rules for ALL:18 AML: 0 -> class ALL   By rules ALL: 1 2 3 4 5 8 9 10 11 13 14 15 16 18 20 21 25 30
Vector of ALL: 18 Recognized by rules for ALL:12 AML: 0 -> class ALL   By rules ALL: 1 15 16 18 19 20 21 23 25 29 30 31
Vector of ALL: 19 Recognized by rules for ALL:18 AML: 0 -> class ALL   By rules ALL: 1 3 4 5 7 8 9 10 11 14 15 16 19 20 21 23 25 31
Vector of ALL: 20 Recognized by rules for ALL:24 AML: 0 -> class ALL   By rules ALL: 1 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Vector of ALL: 21 Recognized by rules for ALL:14 AML: 0 -> class ALL   By rules ALL: 1 3 5 6 7 8 10 12 14 22 23 27 28 31
Vector of ALL: 22 Recognized by rules for ALL:20 AML: 0 -> class ALL   By rules ALL: 2 3 4 5 6 8 9 10 11 12 14 15 16 17 20 21 22 24 25 28
Vector of ALL: 23 Recognized by rules for ALL:30 AML: 0 -> class ALL   By rules ALL: 1 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 - 31
Vector of ALL: 24 Recognized by rules for ALL:25 AML: 0 -> class ALL   By rules ALL: 1 3 4 5 6 7 8 9 10 11 12 14 15 16 17 19 20 21 22 23 24 25 27 28 31
Vector of ALL: 25 Recognized by rules for ALL:25 AML: 0 -> class ALL   By rules ALL: 1 3 4 5 6 7 8 9 10 11 12 14 15 16 17 19 20 21 22 23 24 25 27 28 31
Vector of ALL: 26 Recognized by rules for ALL:18 AML: 0 -> class ALL   By rules ALL: 1 2 3 4 5 8 9 10 11 13 14 15 16 18 20 21 25 30
Vector of ALL: 27 Recognized by rules for ALL:17 AML: 0 -> class ALL   By rules ALL: 1 3 4 5 8 9 10 11 13 14 15 16 18 20 21 25 30
 Recognized true 27, error 0, unknown 0 of 27 vectors of ALL
```

Abbildung 5.4. Die *DataControl* Log-Datei. (Ergebnisse für Lerndatensatz) Der Bericht fängt mit der Serviceinformation (die Zeilen mit "#" am Anfang) an. Jeder der Objekten (Patienten) fängt mit„Vector of " + Klassenname + „Recognized by rules for" an, dann folgt die Information über das Erkennen des Objektes dürch die Regeln von allen Klassen. Z.B. „Vector of ALL: 12 Recognized by rules for ALL:4 AML: 0 -> class ALL By rules ALL: 15 16 18 21" : Objekt 12 wird durch 4 Regeln der Klasse ALL(15 16 18 21) und keine Regeln der Klasse AML erkannt. Objekt wird als ein Objekt der Klasse ALL klassifiziert. Die Information über jede Klasse endet mit der zusammenfassenden Information über die Klasse.

5.1.3. Schlüsselwörterextraktion, Schlüsselwörter und Regelreduktion

Zuerst werden Schlüsselwörter generiert (Tabelle 5.4). In der Tabelle sind für 10 ausgewählte Proteine die generierten Schlüsselwörter dargestellt. Das angewandte Suchprinzip besteht darin, dass das Schlüsselwort für mindestens zwei Proteine gültig ist.

Dann werden die in den logischen Regeln die Merkmale durch die Schlüsselwörter ersetzt(Tabelle 5.5). Die logische Regeln und Schlüsselwörter werden entsprechend der beschriebenen Technik gelöscht. In Abb. 5.5 sind die gelöschten Regeln dargestellt. Zum Beispiel trifft sich das Schlüsselwort «*TRANSMEMBRANE*» in der Regel 2 für *ALL* im Wert "h" und «l». Man kann vermuten, dass in diesem Fall das Schlüsselwort zufällig oder falsch ist. Das vorliegende Schlüsselwort kann die verschiedenen Erkrankungen

nicht unterscheiden und wird gelöscht. Das Löschen dieses Schlüsselwortes wirkt sich nicht nachteilig auf die Klassifikationsgüte der Regelbasis aus.

#	Schlüsselwörter	ATP6C	Azurocidin	Cyclin D3	C-myb	Cystatin C	MCL1	Macmarcks	p62	IOTA	TOP2B
1	CELL	+		+	+		+				
2	HYDROLASE	+								+	
3	EC 3	+								+	
4	ARE PRODUCED BY ALTERNATIVE SPLICING				+		+				+
5	NUCLEAR				+					+	+
6	PHOSPHORYLATION							+			+
7	DIFFERENTIATION				+		+				
8	3D-STRUCTURE		+			+					
9	BELONGS TO PEPTIDASE FAMILY		+							+	
10	ATP	+								+	+
11	TRANSMEMBRANE	+					+				
12	SIGNAL		+			+		+			
13	MACROPHAGE						+	+			
14	NUCLEAR PROTEIN				+						+
15	MONOCYTE		+				+				
16	PROLIFERATION				+		+				
17	SERINE		+	+							
18	CYTOPLASMIC	+	+							+	
19	PROTEASE		+			+				+	

Tabelle 5.4. Die gewählten Schlüsselwörter.

Klasse	Regeln	Merkmale	CELL	HYDROLASE	ARE PRODUCED BY ALTERNATIVE SPLICING	EC 3	DIFFERENTIATION	BELONGS TO PEPTIDASE FAMILY	ATP	TRANSMEMBRANE	SIGNAL	MACROPHAGE	PROLIFERATION	SERINE	CYTOPLASMIC	PROTEASE
ALL	2r 1[H] 6[L]	1[H]ATP6C	H	H		H			H	H					H	
		6[L] MCL1	L		L		L			L		L	L			
	7r 2[L] 1[L]	2[L]Azurocidin						L			L			L	L	L
		1[L]ATP6C	L	L		L			L	L					L	
	17r 3[H] 8[L]	3[H] Cyclin D3	H											H		
		8[L] p62														
	19r 3[H] 1[L]	3[H]Cyclin D3	H											H		
		1[L]ATP6C	L	L		L			L	L					L	
	23r 10[H] 1[L]	10[H]TOP2B			H				H							
		1[L]ATP6C	L	L		L			L	L					L	
	27r 8[L] 1[L]	8[L] p62														
		1[L]ATP6C	L	L		L			L	L					L	
	29r 5[L] 1[L]	5[L] Cystatin C									L					L
		1[L]ATP6C	L	L		L			L	L					L	
	31r 1[L] 4[H]	1[L]ATP6C	L	L		L			L	L					L	
		4[H]C-myb	H		H		H						H			
AML	5r 1[H] 10[L]	1[H]ATP6C	H	H		H			H	H					H	
		10[L]TOP2B			L				L							
	6r 1[H] 3[L]	1[H]ATP6C	H	H		H			H	H					H	
		3[L]Cyclin D3	L											L		
	7r 1[H] 7[L]	1[H]ATP6C	H	H		H			H	H					H	
		7[L]Macmarcks									L	L				
	17r 3[L] 8[H]	3[L] Cyclin D3	L											L		
		8[H] p62														

Tabelle 5.5. Entfernte Regeln

Die Vereinfachung der Regelbasis wird nach der Methode der *DNF*-Erzeugung durchgeführt. In der Tabelle 5.6 sind die Gründe für die Vereinfachung der logischen Regeln dargestellt. Nach der Vereinfachung der Regelbasis bleiben 6 Regeln für die Klasse *ALL* (1, 9, 16, 18, 24, 26) und 4 Regeln für die Klasse *AML* (1, 3, 15, 20) übrig. In Abb. 5.5 sind die Erkennungsergebnisse der Objekte des Kontrolldatensatzes dargestellt. Nur zwei Regeln (1 und 4 für *ALL*) erkennen die Objekte der Klasse *AML* falsch. Die extrahierten 10 logischen Regeln können alle Objekte in den beiden Klassen richtig erkennen (Abb. 5.5 und Tabelle 5.23). Die logischen Regeln sind in der Tabelle 5.7 dargestellt.

Klasse	Warum ist entfernt?	Regeln	Namen	NUCLEAR	PHOSPHORYLATION	3D-STRUCTURE	NUCLEAR PROTEIN	MONOCYTE
ALL		-->>1r 9[H]		+				
	9	***-->>3r 2[L] 6[L]***	***2[L]Azurocidin 6[L]MCL1***			+		+
	9	***-->>4r 2[L] 3[H]***	***2[L]Azurocidin 3[H]Cyclin D3***			+		+
	9	***-->>5r 2[L] 10[H]***	***2[L]Azurocidin 10[H]TOP2B***	+	+	+	+	+
	9	***-->>6r 2[L] 8[L]***	***2[L]Azurocidin 8[L]p62***			+		+
	9	***-->>8r 2[L] 4[H]***	***2[L]Azurocidin 4[H]C-myb***	+		+	+	+
		-->>9r 6[L] 3[H]	6[L]MCL1 3[H]Cyclin D3					+
	9	***-->>10r 6[L] 10[H]***	***6[L]MCL1 10[H]TOP2B***	+	+		+	+
	9	***-->>11r 6[L] 7[H]***	***6[L]MCL1 7[H]Macmarcks***		+			+
	9	***-->>12r 6[L] 8[L]***	***6[L]MCL1 8[L]p62***					+
	9	***-->>13r 6[L] 5[L]***	***6[L]MCL1 5[L]Cystatin C***			+		+
	9	***-->>14r 6[L] 4[H]***	***6[L]MCL1 4[H]C-myb***	+			+	+
	1	***-->>15r 3[H] 10[H]***	***3[H]Cyclin D3 10[H]TOP2B***	+	+		+	
		-->>16r 3[H] 7[H]	3[H]Cyclin D3 7[H]Macmarcks		+			
		-->>18r 3[H] 5[L]	3[H]Cyclin D3 5[L]Cystatin C			+		
	1	***-->>20r 3[H] 4[H]***	***3[H]Cyclin D3 4[H]C-myb***	+			+	
	1	***-->>21r 10[H] 7[H]***	***10[H]TOP2B 7[H]Macmarcks***	+	+		+	
	1	***-->>22r 10[H] 8[L]***	***10[H]TOP2B 8[L]p62***	+	+		+	
		-->>24r 7[H] 8[L]	7[H]Macmarcks 8[L]p62		+			
	1	***-->>25r 7[H] 4[H]***	***7[H]Macmarcks 4[H]C-myb***	+	+		+	
		-->>26r 8[L] 5[L]	8[L]p62 5[L]Cystatin C			+		
	1	***-->>28r 8[L] 4[H]***	***8[L]p62 4[H]C-myb***	+			+	
	26	***-->>30r 5[L] 4[H]***	***5[L]Cystatin C 4[H]C-myb***	+		+	+	
AML		-->>1r 5[H]	5[H]Cystatin C			+		
	3	***-->>2r 9[L] 10[L]***	***9[L]IOTA 10[L]TOP2B***	+	+		+	
		-->>3r 9[L] 3[L]	9[L] IOTA 3[L]Cyclin D3	+				
	3	***-->>4r 9[L] 7[L]***	***9[L] IOTA 7[L]Macmarcks***	+	+			
	1	***-->>8r 10[L] 2[H]***	***10[L]TOP2B 2[H]Azurocidin***	+	+	+	+	+
	3	***-->>9r 10[L] 3[L]***	***10[L]TOP2B 3[L]Cyclin D3***	+	+		+	
	3	***-->>10r 10[L] 6[H]***	***10[L]TOP2B 6[H]MCL1***	+	+		+	+
	3	***-->>11r 10[L] 7[L]***	***10[L]TOP2B 7[L]Macmarcks***	+	+		+	
	3	***-->>12r 10[L] 4[L]***	***10[L]TOP2B 4[L]C-myb***	+	+		+	
	1	***-->>13r 2[H] 3[L]***	***2[H]Azurocidin 3[L]Cyclin D3***			+		+
	1	***-->>14r 2[H] 7[L]***	***2[H]Azurocidin 7[L]Macmarcks***		+	+		+
		-->>15r 3[L] 6[H]	3[L]Cyclin D3 6[H]MCL1					+
	3	***-->>16r 3[L] 4[L]***	***3[L]Cyclin D3 4[L]C-myb***	+			+	
	20	***-->>18r 6[H] 7[L]***	***6[H]MCL1 7[L]Macmarcks***		+			+
	3	***-->>19r 7[L] 4[L]***	***7[L]Macmarcks 4[L]C-myb***	+	+		+	
		-->>20r 7[L] 8[H]	7[L]Macmarcks 8[H]p62		+			

Tabelle 5.6. *DNF*- Erzeugung. Fettmarkierte Regeln sind gelöscht.

```
#Knowledge base name: part4
#Daten base name: data_set_all_aml_independent.dat
Vector of AML: 52 Recognized by rules for AML:3 ALL: 0 -> class AML          By rules AML: 1 3 20
Vector of AML: 53 Recognized by rules for AML:3 ALL: 0 -> class AML          By rules AML: 1 3 15
Vector of AML: 51 Recognized by rules for AML:4 ALL: 0 -> class AML          By rules AML: 1 3 15 20
Vector of AML: 50 Recognized by rules for AML:4 ALL: 0 -> class AML          By rules AML: 1 3 15 20
Vector of AML: 54 Recognized by rules for AML:3 ALL: 0 -> class AML          By rules AML: 1 3 20
Vector of AML: 57 Recognized by rules for AML:3 ALL: 0 -> class AML          By rules AML: 1 3 20
Vector of AML: 58 Recognized by rules for AML:3 ALL: 1 -> class AML          By rules AML: 1 3 15   By rules ALL: 24
Vector of AML: 60 Recognized by rules for AML:2 ALL: 0 -> class AML          By rules AML: 1 3
Vector of AML: 61 Recognized by rules for AML:2 ALL: 0 -> class AML          By rules AML: 3 15
Vector of AML: 65 Recognized by rules for AML:4 ALL: 0 -> class AML          By rules AML: 1 3 15 20
Vector of AML: 66 Recognized by rules for AML:2 ALL: 1 -> class AML          By rules AML: 1 15       By rules ALL: 1
Vector of AML: 63 Recognized by rules for AML:4 ALL: 0 -> class AML          By rules AML: 1 3 15 20
Vector of AML: 64 Recognized by rules for AML:4 ALL: 0 -> class AML          By rules AML: 1 3 15 20
Vector of AML: 62 Recognized by rules for AML:4 ALL: 0 -> class AML          By rules AML: 1 3 15 20
 Recognized true 14, error 0, unk 0 of 14 vectors of AML
Vector of ALL: 39 Recognized by rules for ALL:4 AML: 0 -> class ALL          By rules ALL: 1 9 16 24
Vector of ALL: 40 Recognized by rules for ALL:3 AML: 0 -> class ALL          By rules ALL: 1 24 26
Vector of ALL: 42 Recognized by rules for ALL:4 AML: 0 -> class ALL          By rules ALL: 1 9 16 18
Vector of ALL: 47 Recognized by rules for ALL:1 AML: 0 -> class ALL          By rules ALL: 1
Vector of ALL: 48 Recognized by rules for ALL:4 AML: 0 -> class ALL          By rules ALL: 1 9 16 24
Vector of ALL: 49 Recognized by rules for ALL:4 AML: 0 -> class ALL          By rules ALL: 1 9 16 18
Vector of ALL: 41 Recognized by rules for ALL:4 AML: 0 -> class ALL          By rules ALL: 1 9 16 24
Vector of ALL: 43 Recognized by rules for ALL:4 AML: 0 -> class ALL          By rules ALL: 1 9 16 24
Vector of ALL: 44 Recognized by rules for ALL:6 AML: 0 -> class ALL          By rules ALL: 1 9 16 18 24 26
Vector of ALL: 45 Recognized by rules for ALL:5 AML: 0 -> class ALL          By rules ALL: 1 16 18 24 26
Vector of ALL: 46 Recognized by rules for ALL:5 AML: 0 -> class ALL          By rules ALL: 1 16 18 24 26
Vector of ALL: 70 Recognized by rules for ALL:5 AML: 0 -> class ALL          By rules ALL: 1 16 18 24 26
Vector of ALL: 71 Recognized by rules for ALL:1 AML: 0 -> class ALL          By rules ALL: 16
Vector of ALL: 72 Recognized by rules for ALL:3 AML: 0 -> class ALL          By rules ALL: 1 16 18
Vector of ALL: 68 Recognized by rules for ALL:3 AML: 0 -> class ALL          By rules ALL: 1 16 18
Vector of ALL: 69 Recognized by rules for ALL:3 AML: 0 -> class ALL          By rules ALL: 1 9 16
Vector of ALL: 67 Recognized by rules for ALL:2 AML: 0 -> class ALL          By rules ALL: 16 18
Vector of ALL: 55 Recognized by rules for ALL:6 AML: 0 -> class ALL          By rules ALL: 1 9 16 18 24 26
Vector of ALL: 56 Recognized by rules for ALL:5 AML: 0 -> class ALL          By rules ALL: 9 16 18 24 26
Vector of ALL: 59 Recognized by rules for ALL:6 AML: 0 -> class ALL          By rules ALL: 1 9 16 18 24 26
 Recognized true 20, error 0, unk 0 of 20 vectors of ALL
```

Abbildung 5.5. *DataControl* log Datei für die neuen Regeln (Ergebnisse für Kontrolldatensatz). Beschreibung s. Abb. 5.4.

In der Tabelle 5.8 ist die Klassifikation der Leukämieformen dargestellt. Man kann bemerken, dass die monozytische Form der Leukämie durch das Schlüsselwort («MONOCYTE») bestimmt wird. Die Analyse erbrachte das Ergebnis, das die Schlüsselwörter «NUCLEAR», «PHOSPHORYLATION und «3D-STRUCTURE» sich bei den verschiedenen Formen stark unterscheiden.

Klasse	Regel #	Regelbeschreibug	NUCLEAR	PHOSPHORY-LATION	3D-STRUCTURE	MONOCYTE	Λ (in Lerndaten-satz)	Λ (in Kontrollda-tensatz)
ALL	1	9[H] IOTA	+				24 (88%)	17(85%)
	9	6[L]MCL1 3[H]Cyclin D3				+	23 (95%)	11(55%)
	16	3[H]Cyclin D3 7[H]Macmarcks		+			25 (92%)	18(90%)
	18	3[H]Cyclin D3 5[L]Cystatin C			+		16 (59%)	12(60%)
	24	7[H]Macmarcks 8[L]p62		+			16(59%)	12(60%)
	26	8[L]p62 5[L]Cystatin C			+		8(29%)	8(40%)
AML	1	5[H]Cystatin C			+		9 (81%)	13(92%)
	3	9[L] IOTA 3[L]Cyclin D3	+				10 (90%)	13(92%)
	15	3[L]Cyclin D3 6[H]MCL1				+	9 (81%)	10(71%)
	20	7[L]Macmarcks 8[H]p62		+			7 (63%)	9(64%)

Tabelle 5.7. Finalversion von Regeln für Leukämie Datensatz

FAB Tradition	Cells	Nucleus	Occurrence
L1 ALL	Small, uniform	Round, homogeneous	75% of childhood ALL
L2 ALL	Of varying size	Irregular, no homogeneous	70% of adult ALL
L3 Burkett-type ALL	Large, uniform	Round-to-oval, homogeneous	Rare
M1 AML without maturity	Large, regular	Round, regular	M1+M2+M3= 45% of adult ANLL
M2 AML with signs of maturity	Very large	Kidney-shaped	
M3 Acute promyleocytic	Very large	Kidney Shaped	
M4 Acute myelomonocytic leukemia	Like M2+M5, each equaling at least 20% in the marrow or peripheral blood	=	40% of adult ANLL
M5 Acute monocytic	large, often lobulated	Indented	7% of adult ANLL
M6 Acute ertyhroleukemia	Large, bizarre, round-to-oval	Large, lobulated round-to-oval	Rare
-- Acute undifferentiated	Often resembling L1, L2,M1	Large, lobulated round-to-oval	Rare

Tabelle 5.8. Klassifikation von akuter Leukämie (http://www.meds.com/leukemia)

5.2. Datensatz „Darmkrebs“

5.2.1. Auswahl der Gene

In der ersten Etappe wurden die Merkmale gewählt, die den verschiedenen Schwellen Sc entsprechen (Abb. 5.6). Die neuen logischen Regeln wurden hinsichtlich der Erkennungsqualität (Anzahl der richtigen Erkennungen Λ im Trainingsatz) verglichen worden. Die Ergebnisse befinden sich in der Tabelle 5.9. Die Erkennungsqualität der logischen Regeln hängt bei Schwelle 40 % und niedriger nicht nicht von der Merkmalanzahl ab, deshalb kann man die Schwelle 60 % verwenden. Die Anzahl der Merkmale ist dann 11.

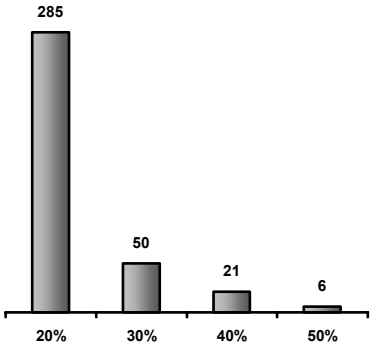

Abbildung 5.6 Die Merkmalanzahl für verschiedene Schwellen Sc

Sc	Merkmalanzahl	Regelanzahl (T/N)	Anzahl der richtigen Erkennungen Λ
20%	35	285 (187/98)	45 von 45
30%	15	50 (30/20)	45 von 45
40%	11	21(12/9)	45 von 45
50%	6	-	45 von 45

Tabelle 5.9. Die Erkennungsqualität je nach der Schwelle Sc

Die ausgewählten Gene befinden sich in der Tabelle 5.10 und die graphische Darstellung der gewählten Gene ist in Abbildung 5.7 gezeigt.

In diesem Datensatz kann man deutliche Unterschiede zwischen den Genen erkennen. Es existieren drei Gruppen von Genen: niedrigexprimierte und hochexprimierte bei Krebs und hochexprimierte ohne Krebs. Die ausgewählten Gene wurden durch andere Autoren nicht gefunden. Nach der nochmaligen Kodierung (nur aufgrund der gewählten Gene) ist das Bild nicht so deutlich, wie auf der Abbildung 5.7. Das Problem kann durch die Nutzung der logischen Regeln gelöst werden.

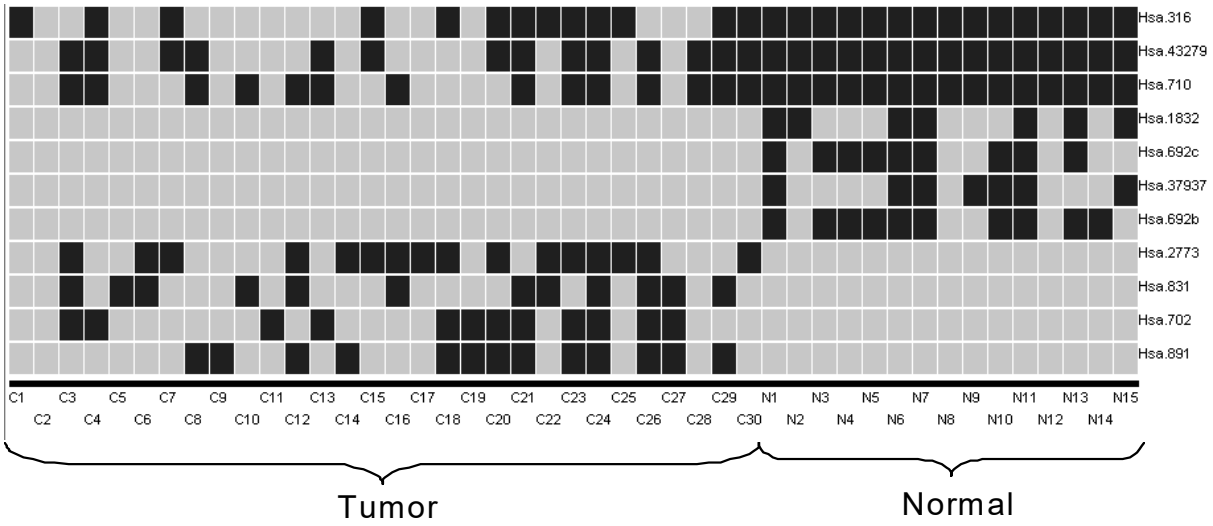

Abbildung 5.7. Graphische Darstellung der ausgewählten Gene (GeneRule 2.0). Die hochexprimierte Gene sind schwarz.

Gene	Swiss-Prot/TrEMBL Namen	Beschreibung
Hsa.2773	HMG1_HUMAN	Human mRNA for HMG-1.
Hsa.692 (2)	CYSR_HUMAN	Human cysteine-rich protein (CRP) gene, exons 5 and 6.
Hsa.43279	CD37_HUMAN	LEUKOCYTE ANTIGEN CD37 (Homo sapiens)
Hsa.891	TYY1_HUMA	Human lysozyme mRNA, complete cds.
Hsa.702	PPIB_HUMAN	PEPTIDYL-PROLYL CIS-TRANS ISOMERASE B PRECURSOR (HUMAN)
Hsa.37937	MYH9_HUMAN	MYOSIN HEAVY CHAIN, NONMUSCLE (Gallus gallus)
Hsa.831	DD10_HUMAN	MITOCHONDRIAL MATRIX PROTEIN P1 PRECURSOR (HUMAN);.
Hsa.316	PRD2_HUMAN	Human mucin 2 (MUC2) mRNA sequence.
Hsa.1832	MLRN_HUMAN	MYOSIN REGULATORY LIGHT CHAIN 2, SMOOTH MUSCLE ISOFORM
Hsa.710	K2C8_HUMAN	KERATIN, TYPE II CYTOSKELETAL 8 (HUMAN).

Tabelle 5.10. Ausgewählte Gene

5.2.2. Die Regelextraktion

In Abb. 5.8 sind die extrahierten Regeln dargestellt. Die 51 gefundenen Regeln können alle Objekte vom Lerndatensatz korrekt erkennen (s. Abb. 5.9).

Die Merkmale 6 [h] und 6[l] kommen in beiden Klassen und können gelöscht werden. Die den Merkmalen entsprechenden Regeln (Krebs 6, 8, 9 und normal 4, 8, 10) werden deshalb auch gelöscht. Die Merkmale 8 und 2 sind gleich (das ist ein Gen Hsa.692 oder CYSR_HUMAN) und deswegen können die Regeln 5 und 9 für die Klasse Normal gelöscht werden (Regel 1 statt 5 und 9).

***Knowledge base name: cc**
***Daten base name: cc_data_transformed_train.dat**
***In Class KREBS: 30, in another Class: 15**
->> 1r 1 [h] in KREBS: 27 (90 %) in non KREBS: 0 (0 %)
->> 2r 3 [l] in KREBS: 6 (20 %) in non KREBS: 0 (0 %)
->> 3r 7 [h] in KREBS: 25 (83 %) in non KREBS: 0 (0 %)
->> 4r 9 [l] in KREBS: 7 (23 %) in non KREBS: 0 (0 %)
->> 5r 5 [h] 2 [l] in KREBS: 24 (80 %) in non KREBS: 0 (0 %)
->> 6r 5 [h] 6 [h] in KREBS: 3 (10 %) in non KREBS: 0 (0 %)
->> 7r 5 [h] 4 [h] in KREBS: 16 (53 %) in non KREBS: 0 (0 %)
->> 8r 2 [l] 6 [l] in KREBS: 24 (80 %) in non KREBS: 0 (0 %)
->> 9r 6 [l] 4 [h] in KREBS: 18 (60 %) in non KREBS: 0 (0 %)

Abbildung 5.8a. Generierte Regeln Klasse Krebs (*GeneRule* log-Datei). Die Beschreibung s. Abb. 5.3.

***Knowledge base name: cc**
***Daten base name: cc_data_transformed_train.dat**
***In Class NORMAL: 15, in another Class: 30**
->> 1r 8 [h] in NORMAL: 10 (66 %) in non NORMAL: 0 (0 %)
->> 2r 3 [h] 10 [h] in NORMAL: 7 (46 %) in non NORMAL: 0 (0 %)
->> 3r 3 [h] 7 [l] 1 [l] in NORMAL: 15 (100 %) in non NORMAL: 0 (0 %)
->> 4r 3 [h] 7 [l] 6 [h] in NORMAL: 8 (53 %) in non NORMAL: 0 (0 %)
->> 5r 1 [l] 2 [h] in NORMAL: 12 (80 %) in non NORMAL: 0 (0 %)
->> 6r 1 [l] 4 [l] in NORMAL: 14 (93 %) in non NORMAL: 0 (0 %)
->> 7r 1 [l] 5 [l] in NORMAL: 14 (93 %) in non NORMAL: 0 (0 %)
->> 8r 2 [h] 6 [h] in NORMAL: 5 (33 %) in non NORMAL: 0 (0 %)
->> 9r 2 [h] 10 [h] in NORMAL: 5 (33 %) in non NORMAL: 0 (0 %)
->> 10r 6 [h] 5 [l] in NORMAL: 8 (53 %) in non NORMAL: 0 (0 %)
->> 11r 10 [h] 4 [l] in NORMAL: 7 (46 %) in non NORMAL: 0 (0 %)
->> 12r 10 [h] 5 [l] in NORMAL: 7 (46 %) in non NORMAL: 0 (0 %)

Abbildung 5.8b. Generierte Regeln Klasse Normal (*GeneRule* log-Datei). Die Beschreibung s. Abb. 5.3.

#Knowledge base name: cc
#Daten base name: cc_data_transformed_train.dat

Vector of normal: N1 Recognized by rules for normal:12 krebs: 0 -> class normal	By rules normal: 1 2 3 4 5 6 7 8 9 10 11 12
Vector of normal: N2 Recognized by rules for normal:9 krebs: 0 -> class normal	By rules normal: 1 2 3 4 6 7 10 11 12
Vector of normal: N3 Recognized by rules for normal:5 krebs: 0 -> class normal	By rules normal: 1 3 5 6 7
Vector of normal: N4 Recognized by rules for normal:5 krebs: 0 -> class normal	By rules normal: 1 3 5 6 7
Vector of normal: N5 Recognized by rules for normal:5 krebs: 0 -> class normal	By rules normal: 1 3 5 6 7
Vector of normal: N6 Recognized by rules for normal:12 krebs: 0 -> class normal	By rules normal: 1 2 3 4 5 6 7 8 9 10 11 12
Vector of normal: N7 Recognized by rules for normal:12 krebs: 0 -> class normal	By rules normal: 1 2 3 4 5 6 7 8 9 10 11 12
Vector of normal: N8 Recognized by rules for normal:3 krebs: 0 -> class normal	By rules normal: 3 5 6
Vector of normal: N9 Recognized by rules for normal:4 krebs: 0 -> class normal	By rules normal: 3 4 7 10
Vector of normal: N10 Recognized by rules for normal:5 krebs: 0 -> class normal	By rules normal: 1 3 5 6 7
Vector of normal: N11 Recognized by rules for normal:12 krebs: 0 -> class normal	By rules normal: 1 2 3 4 5 6 7 8 9 10 11 12
Vector of normal: N12 Recognized by rules for normal:7 krebs: 0 -> class normal	By rules normal: 3 4 5 6 7 8 10
Vector of normal: N13 Recognized by rules for normal:9 krebs: 0 -> class normal	By rules normal: 1 2 3 5 6 7 9 11 12
Vector of normal: N14 Recognized by rules for normal:4 krebs: 0 -> class normal	By rules normal: 3 5 6 7
Vector of normal: N15 Recognized by rules for normal:8 krebs: 0 -> class normal	By rules normal: 2 3 4 6 7 10 11 12
Recognized true 15, error 0, unk 0 of 15 vectors of normal	
Vector of krebs: C1 Recognized by rules for krebs:4 normal: 0 -> class krebs	By rules krebs: 1 3 5 6
Vector of krebs: C2 Recognized by rules for krebs:4 normal: 0 -> class krebs	By rules krebs: 2 5 6 7
Vector of krebs: C3 Recognized by rules for krebs:4 normal: 0 -> class krebs	By rules krebs: 1 3 5 8
Vector of krebs: C4 Recognized by rules for krebs:4 normal: 0 -> class krebs	By rules krebs: 1 3 5 8
Vector of krebs: C5 Recognized by rules for krebs:7 normal: 0 -> class krebs	By rules krebs: 1 2 3 5 7 8 9
Vector of krebs: C6 Recognized by rules for krebs:4 normal: 0 -> class krebs	By rules krebs: 1 3 5 8
Vector of krebs: C7 Recognized by rules for krebs:2 normal: 0 -> class krebs	By rules krebs: 1 3
Vector of krebs: C8 Recognized by rules for krebs:6 normal: 0 -> class krebs	By rules krebs: 3 4 5 7 8 9
Vector of krebs: C9 Recognized by rules for krebs:8 normal: 0 -> class krebs	By rules krebs: 1 2 3 4 5 7 8 9
Vector of krebs: C10 Recognized by rules for krebs:5 normal: 0 -> class krebs	By rules krebs: 1 3 4 5 8
Vector of krebs: C11 Recognized by rules for krebs:4 normal: 0 -> class krebs	By rules krebs: 1 3 5 6
Vector of krebs: C12 Recognized by rules for krebs:7 normal: 0 -> class krebs	By rules krebs: 1 3 4 5 7 8 9
Vector of krebs: C13 Recognized by rules for krebs:6 normal: 0 -> class krebs	By rules krebs: 1 3 5 7 8 9
Vector of krebs: C14 Recognized by rules for krebs:8 normal: 0 -> class krebs	By rules krebs: 1 2 3 4 5 7 8 9
Vector of krebs: C15 Recognized by rules for krebs:3 normal: 0 -> class krebs	By rules krebs: 1 5 8
Vector of krebs: C16 Recognized by rules for krebs:4 normal: 0 -> class krebs	By rules krebs: 1 3 8 9
Vector of krebs: C17 Recognized by rules for krebs:4 normal: 0 -> class krebs	By rules krebs: 1 3 5 8
Vector of krebs: C18 Recognized by rules for krebs:6 normal: 0 -> class krebs	By rules krebs: 1 3 5 7 8 9
Vector of krebs: C19 Recognized by rules for krebs:5 normal: 0 -> class krebs	By rules krebs: 1 2 4 7 9
Vector of krebs: C20 Recognized by rules for krebs:5 normal: 0 -> class krebs	By rules krebs: 1 5 7 8 9
Vector of krebs: C21 Recognized by rules for krebs:5 normal: 0 -> class krebs	By rules krebs: 3 5 7 8 9
Vector of krebs: C22 Recognized by rules for krebs:4 normal: 0 -> class krebs	By rules krebs: 1 3 5 8
Vector of krebs: C23 Recognized by rules for krebs:6 normal: 0 -> class krebs	By rules krebs: 1 3 5 7 8 9
Vector of krebs: C24 Recognized by rules for krebs:6 normal: 0 -> class krebs	By rules krebs: 1 3 5 7 8 9
Vector of krebs: C25 Recognized by rules for krebs:3 normal: 0 -> class krebs	By rules krebs: 1 3 8
Vector of krebs: C26 Recognized by rules for krebs:7 normal: 0 -> class krebs	By rules krebs: 1 3 4 5 7 8 9
Vector of krebs: C27 Recognized by rules for krebs:7 normal: 0 -> class krebs	By rules krebs: 1 2 3 5 7 8 9
Vector of krebs: C28 Recognized by rules for krebs:6 normal: 0 -> class krebs	By rules krebs: 1 3 5 7 8 9
Vector of krebs: C29 Recognized by rules for krebs:4 normal: 0 -> class krebs	By rules krebs: 1 3 8 9
Vector of krebs: C30 Recognized by rules for krebs:2 normal: 0 -> class krebs	By rules krebs: 1 9
Recognized true 30, error 0, unk 0 of 30 vectors of krebs	

Abbildung 5.9. Ein Teil der *DataControl* log-Datei für die neuen Regeln (Ergebnisse für Lerndatensatz). Beschreibung s. Abb. 5.4.

Die erzeugten logischen Regeln sind *DNF*, deshalb ist eine weitere Vereinfachung nicht möglich. Die Nutzung der Expertenanalyse ist auch nicht möglich. Die Nutzung der automatischen Clusterbildung hat keine befriedigenden Ergebnisse gebracht, da keine stabilen Cluster gefunden werden konnten. Deswegen ist eine befriedigende Lösung mit Hilfe der Clustering nicht möglich. Im nächsten Abschnitt wird die Möglichkeit der Anwendung des logischen Herangehens nach Substitution durch Schlüsselwörter untersucht.

.

5.2.3. Schlüsselwörterextraktion, Schlüsselwörter und Regelreduktion

Zuerst werden Schlüsselwörter generiert. In der Tabelle 5.11 sind für 11 ausgewählte Proteine die generierten Schlüsselwörter dargestellt. Das Suchprinzip besteht darin, dass ein Schlüsselwort für mindestens zwei Proteine gültig ist. Dann werden in den logischen Regeln die Merkmale durch Schlüsselwörter ersetzt (s. Tab. 5.12).

#	Schluesselwörter	HMG1	CYSR	CD37	TYY1	PPIB	MYH9	DD10	CYSR	PRD2	MLRN	K2C8
1	BINDING;ZINC		+						+	+		
2	TRANSCRIPTION REGULATION				+					+		
3	NUCLEAR		+		+				+	+		
4	COIL						+					+
5	ACTIN						+				+	
6	PHOSPHORYLATION										+	+
7	ATP-BINDING						+	+				
9	METAL-BINDING		+		+				+	+		
10	NUCLEAR PROTEIN;REPEAT		+						+	+		
11	MYOSIN						+				+	
12	MLC						+				+	
13	NUCLEAR PROTEIN		+		+				+	+		
14	ZINC		+		+				+	+		
15	ZINC-FINGER;METAL-BINDING				+					+		

Tabelle 5.11. Die gewählten Schlüsselwörter.

. Klasse	Regel #	Merkmale	Beschreibung	NUCLEAR	METAL-BINDING	NUCLEAR PROTEIN;REPEAT	NUCLEAR PROTEIN	ZINC	ZINC-FINGER;METAL-BINDING
Krebs	4	9[l]	Hsa.316	L	L	L	L	L	L
	5	5[h]	Hsa.702						
		2[l]	Hsa.692b	L	L	L	L	L	
	7	5[h]	Hsa.702						
		4[h]	Hsa.891	H	H		H	H	H
Normal	1	8[h]	Hsa.692c	H	H	H	H	H	
	6	1[l]	Hsa.2773						
		4[l]	Hsa.891	L	L		L	L	L
	1	10[h]	Hsa.1832						
	1	4[l]	Hsa.891	L	L		L	L	L

Tabelle 5.12. Entfernte Regeln

Klasse	Regel #	Merkmale	Beschreibung	BINDING;ZINC	TRANSCRIPTION REGULATION	COIL	ACTIN	PHOSPHORYLATION	ATP-BINDING	MYOSIN	MLC
Krebs	1	1[h]	Hsa.2773								
	2	3[l]	Hsa.43279								
	3	7[h]	Hsa.831						H		
Normal	2	3[h]	Hsa.43279								
		10[h]	Hsa.1832				H	H		H	H
	3	3[h]	Hsa.43279								
		7[l]	Hsa.831						L		
		1[l]	Hsa.2773								
	7	1[l]	Hsa.2773								
		5[l]	Hsa.702								
	12	10[h]	Hsa.1832				H	H		H	H
		5[l]	Hsa.702								

Tabelle 5.13. Regeln nach der Reduktion.

Am Ende der Vereinfachung bleiben 3 Regeln für die Klasse Krebs (1, 2, 7) und 4 Regeln für die Klasse Normal (2, 3, 7, 12) übrig. Eine weitere Vereinfachung durch die logischen Methoden ist nicht möglich (s. Tab.5.13).

```
#Knowledge base name: cc
#Daten base name: cc_data_transformed_indep.dat
Vector of normal: N16 Recognized by rules for normal:2 krebs: 1 -> class normal   By rules normal: 2 12     By rules krebs: 1
Vector of normal: N17 Recognized by rules for normal:2 krebs: 1 -> class normal   By rules normal: 7 12     By rules krebs: 2
Vector of normal: N18 Recognized by rules for normal:0 krebs: 2 -> class krebs                               By rules krebs: 1 3      ERROR
Vector of normal: N19 Recognized by rules for normal:4 krebs: 0 -> class normal   By rules normal: 2 3 7 12
Vector of normal: N20 Recognized by rules for normal:0 krebs: 3 -> class krebs                               By rules krebs: 1 2 3    ERROR
Vector of normal: N21 Recognized by rules for normal:2 krebs: 1 -> class normal   By rules normal: 2 12     By rules krebs: 1
Vector of normal: N22 Recognized by rules for normal:4 krebs: 0 -> class normal   By rules normal: 2 3 7 12
 Recognized true 5, error 2, unk 0 of 7 vectors of normal
Vector of krebs: C31 Recognized by rules for krebs:2 normal: 0 -> class krebs     By rules krebs: 1 3
Vector of krebs: C32 Recognized by rules for krebs:2 normal: 0 -> class krebs     By rules krebs: 1 3
Vector of krebs: C33 Recognized by rules for krebs:2 normal: 1 -> class krebs     By rules normal: 12      By rules krebs: 1 2
Vector of krebs: C34 Recognized by rules for krebs:2 normal: 0 -> class krebs     By rules krebs: 1 3
Vector of krebs: C35 Recognized by rules for krebs:1 normal: 0 -> class krebs     By rules krebs: 1
Vector of krebs: C36 Recognized by rules for krebs:0 normal: 2 -> class normal    By rules normal: 3 7                      ERROR
Vector of krebs: C37 Recognized by rules for krebs:3 normal: 0 -> class krebs     By rules krebs: 1 2 3
Vector of krebs: C38 Recognized by rules for krebs:2 normal: 0 -> class krebs     By rules krebs: 1 3
Vector of krebs: C39 Recognized by rules for krebs:2 normal: 0 -> class krebs     By rules krebs: 1 3
Vector of krebs: C40 Recognized by rules for krebs:3 normal: 0 -> class krebs     By rules krebs: 1
Recognized true 9, error 1, unk 0 of 10 vectors of krebs
```

Abbildung 5.10. *DataControl* log Datei für die neuen Regeln (Ergebnisse für Kontrolldatensatz). Beschreibung s. Abb. 5.4.

In Abb. 5.10 ist eine Protokoll-Datei des Programms *DataControl* dargestellt. Die 7 gefundenen Regeln können 14 von 17 Objekten des Kontrolldatensatzes korrekt erkennen (s. Tab. 5.14). Die erhaltenen Schlüsselwörter "Actin", "Myosin" und "MLC" sind Bestandteile eines Muskelgewebes und kommen im Tumorgewebe nicht vor. Das bestätigt die Richtigkeit der gefundenen Regeln.

Klasse	Regel #	Beschreibung	ACTIN	PHOSPHORYLATION	ATP-BINDING	MYOSIN	MLC	Λ IN TRAINING SATZ	Λ IN INDEPENDENT. SATZ	FEHLER IN INDEPEN-DENT. SATZ
Krebs	1	1[H]Hsa.2773						27(90%)	9(90%)	4(57%)
	2	3[L]Hsa.43279						6(20%)	3(30%)	2(28%)
	3	7[H]Hsa.831			H			25(83%)	7(70%)	2(28%)
Normal	2	3[H]Hsa.43279 and 10[H]Hsa.1832	H	H		H	H	7(46%)	4(57%)	
	3	3[H]Hsa.43279 and 7[L]Hsa.831 and 1[L]Hsa.2773			L			15(100%)	2(28%)	1(10%)
	7	1[L]Hsa.2773 and 5[L]Hsa.702						14(93%)	3(42%)	1(10%)
	12	10[H]Hsa.1832 and 5[L]Hsa.702	H	H		H	H	7(56%)	5(71%)	1(10%)

Tabelle 5.14. Finalversion der Regeln für Darmkrebs Datensatz

5.3. Datensatz „Hirntumor“

5.3.1. Auswahl der Gene

In der ersten Etappe wurden die Merkmale gewählt, die den verschiedenen Schwellen Sc entsprechen (Abb. 5.11). Die neuen logischen Regeln wurden hinsichtlich der Erkennungsqualität (Anzahl der richtigen Erkennungen Λ im Trainingsatz) verglichen worden. Die Ergebnisse befinden sich in der Tabelle 5.15. Die Erkennungsqualität der logischen Regeln hängt bei Schwelle 20 % und niedriger nicht von der Merkmalanzahl ab, deshalb kann man die Schwelle 20 % verwenden. Die Anzahl der Merkmale ist dann 20. Die ausgewählten Gene befinden sich in der Tabelle 5.16 und die graphische Darstellung der gewählten Gene ist in Abbildung 5.12 gezeigt.

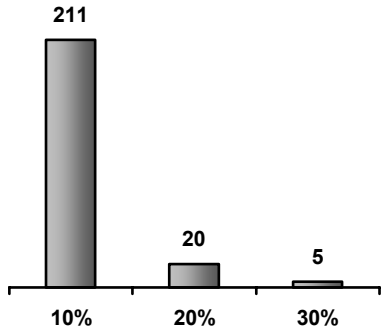

Abbildung 5.11 Die Merkmalanzahl für verschiedene Schwellen Sc

Sc	Merkmalanzahl	Regelanzahl (TF/S)	Anzahl der richtigen Erkennungen Λ
10%	211	42/67	39 of 40
20%	20	14/47	34 of 40
30%	5	-	-

Tabelle 5.15. Die Erkennungsqualität je nach der Schwelle Sc.

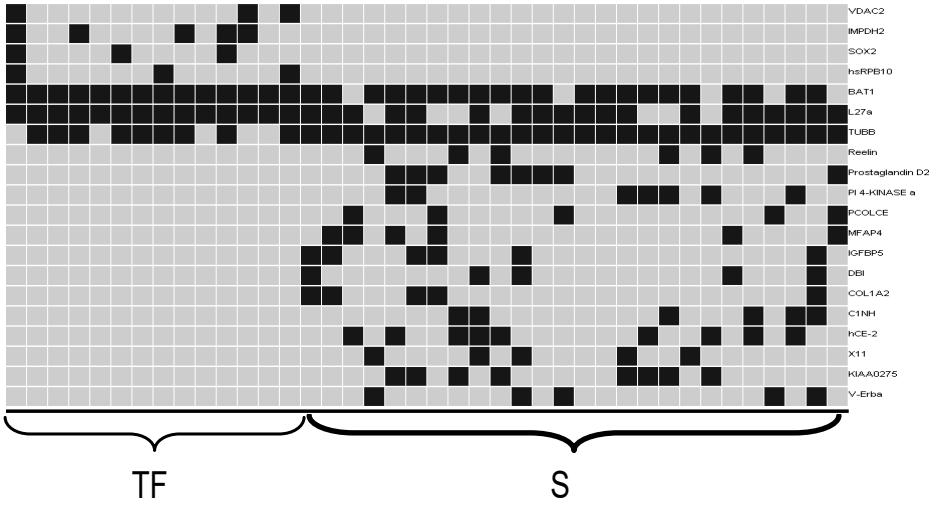

Abbildung 5.12. Die graphische Darstellung der gewählten Gene (screenshot GeneRule 2.0). Die hochexprimierte Gene sind schwarz. (TF – Treatment Failure; S – Survivors)

1	(clone FFE-7) type II inosine monophosphate dehydrogenase (IMPDH2)	IMD2_HUMAN
2	BAT1 mRNA for nuclear RNA helicase (DEAD family)	BAT1_HUMAN
3	C1NH Complement component 1 inhibitor	IC1_HUMAN
4	Carboxylesterase (hCE-2)	Q9UKY3
5	COL1A2 Collagen, type I,	CA21_HUMAN
6	Diazepam binding inhibitor	ACBP_HUMAN
7	Insulin-like growth factor binding protein 5 (IGFBP5)	IBP5_HUMAN
8	KIAA0275 gene	TIC2_HUMAN
9	MFAP4 Microfibrillar-associated protein 4	MFA4_HUMAN
10	PCOLCE Procollagen C-endopeptidase enhancer	PCO1_HUMAN
11	PHOSPHATIDYLINOSITOL 4-KINASE ALPHA	PI4K_HUMAN
12	Prostaglandin D2 synthase gene	P41222
13	Reelin (RELN) mRNA	P78509
14	Ribosomal protein L27a mRNA	RL2A_HUMAN
15	RNA polymerase II subunit (hsRPB10) mRNA	RPBX_HUMAN
16	SOX2 SRY (sex determining region Y)-box 2	SOX2_HUMAN
17	TUBB Beta-tubulin	TBB4_HUMAN
18	VDAC2 Voltage-dependent anion channel 2	POR2_HUMAN
19	V-Erba Related Ear-3 Protein	COT1_HUMAN
20	X11 protein mRNA, partial cds	APB2_HUMAN

Tabelle 5.16. Ausgewählte Gene

In diesem Datensatz ist eine deutliche Grenze zwischen den Genen zu erkennen. Es gibt vier Gruppen von Genen: niedrig- und hochexprimierte bei TF (Treatment failure), niedrig- und hochexprimierte bei Überlebenden S (*survival*). Nach der zweiten Kodierung (nur auf der Basis von selektierten Genen) ist das Bild nicht mehr so deutlich. Das Problem kann durch die Nutzung der logischen Regeln gelöst werden.

5.3.2. Die Regelextraktion

In Abb. 5.13 sind die extrahierten Regeln dargestellt. Die 61 gefundenen Regeln können 33 von 40 Objekten vom Lerndatensatz (s. Abb. 5.14). Eine weitere Vereinfachung der Regelbasis durch logische Methoden und Expertenanalyse ist nicht möglich. In Abb. 5.14 sind die Ergebnisse (logische Regeln) des Programms *DataControl* dargestellt. Im nächsten Abschnitt werden die die Möglichkeiten von Clusteranalyse untersucht. Das Erkennungsbild für die Klasse *TF* ist ziemlich kompliziert. Die Nutzung von Clustermethoden wird in diesem Fall zu keinem eindeutigen Ergebnis führen und man kann die Nutzung der Clusteranalyse für die Klasse *TF* ausschließen. Für die Klasse "*survivors*" kann man einige Vereinfachungen

vornehmen. Alle Objekte können in zwei Gruppen aufgeteilt werden. Die erste Gruppe besteht aus den Objekten, die durch die Regeln 1, 6 und 11 erkannt werden. und die zweite Gruppe wird durch die Regeln 2 und 3 erkannt. Die Regeln 4, 5, 7, 8, 9, 10, 12 und14 können gelöscht werden.

```
Regeln TF:
-->>1r 1[H] in TF: 13(50%) in non TF: 0(0%)
-->>2r 5[H] in TF: 7(26%) in non TF: 0(0%)
-->>3r 11[L] in TF: 4(15%) in non TF: 0(0%)
-->>4r 8[H] 12[L] in TF: 8(30%) in non TF: 0(0%)
-->>5r 8[H] 9[H] in TF: 5(19%) in non TF: 0(0%)
-->>6r 8[H] 2[H] in TF: 5(19%) in non TF: 0(0%)
-->>7r 8[H] 3[L] in TF: 7(26%) in non TF: 0(0%)
-->>8r 8[H] 7[L] in TF: 5(19%) in non TF: 0(0%)
-->>9r 16[L] 2[H] in TF: 3(11%) in non TF: 0(0%)
-->>10r 16[L] 6[L] in TF: 3(11%) in non TF: 0(0%)
-->>11r 19[L] 12[L] in TF: 14(53%) in non TF: 0(0%)
-->>12r 19[L] 4[H] 20[H] in TF: 3(11%) in non TF: 0(0%)
-->>13r 19[L] 4[H] 15[L] 2[H] in TF: 4(15%) in non TF: 0(0%)
-->>14r 4[H] 9[H] 14[H] in TF: 3(11%) in non TF: 0(0%)
-->>15r 12[L] 9[L] in TF: 12(46%) in non TF: 0(0%)
-->>16r 12[L] 2[H] in TF: 7(26%) in non TF: 0(0%)
-->>17r 12[L] 20[L] in TF: 7(26%) in non TF: 0(0%)
-->>18r 12[L] 3[H] in TF: 8(30%) in non TF: 0(0%)
-->>19r 12[L] 14[L] in TF: 8(30%) in non TF: 0(0%)
-->>20r 12[L] 14[H] in TF: 6(23%) in non TF: 0(0%)
-->>21r 12[L] 4[L] in TF: 8(30%) in non TF: 0(0%)
-->>22r 12[L] 7[L] in TF: 6(23%) in non TF: 0(0%)
-->>23r 12[L] 15[H] in TF: 6(23%) in non TF: 0(0%)
-->>24r 12[L] 10[H] in TF: 10(38%) in non TF: 0(0%)
-->>25r 12[L] 18[H] in TF: 5(19%) in non TF: 0(0%)
-->>26r 9[H] 10[H] 18[L] in TF: 5(19%) in non TF: 0(0%)
-->>27r 2[H] 14[H] in TF: 5(19%) in non TF: 0(0%)
-->>28r 2[H] 10[H] in TF: 9(34%) in non TF: 0(0%)
-->>29r 2[H] 18[H] in TF: 3(11%) in non TF: 0(0%)
-->>30r 20[L] 18[L] 14[H] in TF: 4(15%) in non TF: 0(0%)
-->>31r 20[L] 18[L] 6[H] in TF: 3(11%) in non TF: 0(0%)
-->>32r 20[L] 18[L] 15[L] in TF: 4(15%) in non TF: 0(0%)
-->>33r 20[L] 18[L] 10[H] in TF: 5(19%) in non TF: 0(0%)
-->>34r 20[H] 15[H] in TF: 5(19%) in non TF: 0(0%)
-->>35r 20[H] 18[H] in TF: 4(15%) in non TF: 0(0%)
-->>36r 20[H] 10[H] 6[L] in TF: 7(26%) in non TF: 0(0%)
-->>37r 3[L] 10[H] 4[L] in TF: 4(15%) in non TF: 0(0%)
-->>38r 3[L] 10[H] 6[L] in TF: 6(23%) in non TF: 0(0%)
-->>39r 3[L] 10[H] 14[H] in TF: 4(15%) in non TF: 0(0%)
-->>40r 6[L] 15[L] 10[H] in TF: 7(26%) in non TF: 0(0%)
-->>41r 6[L] 15[L] 4[L] in TF: 5(19%) in non TF: 0(0%)
-->>42r 6[L] 15[L] 14[L] in TF: 7(26%) in non TF: 0(0%)
-->>43r 6[L] 15[L] 7[L] in TF: 6(23%) in non TF: 0(0%)
-->>44r 14[L] 15[L] 18[H] 7[L] in TF: 3(11%) in non TF: 0(0%)
-->>45r 14[H] 7[L] in TF: 4(15%) in non TF: 0(0%)
-->>46r 14[H] 15[H] in TF: 5(19%) in non TF: 0(0%)
-->>47r 4[L] 10[H] 15[H] in TF: 5(19%) in non TF: 0(0%)

Regeln Survivors
-->>1r 8[L] 19[H] in SURVIVORS: 5(35%) in non SURVIVORS: 0(0%)
-->>2r 8[H] 2[L] 12[H] 3[H] in SURVIVORS: 3(21%) in non SURVIVORS: 0(0%)
-->>3r 8[H] 2[L] 12[H] 9[L] in SURVIVORS: 3(21%) in non SURVIVORS: 0(0%)
-->>4r 19[H] 12[H] in SURVIVORS: 4(28%) in non SURVIVORS: 0(0%)
-->>5r 19[H] 9[L] in SURVIVORS: 4(28%) in non SURVIVORS: 0(0%)
-->>6r 19[H] 18[L] in SURVIVORS: 5(35%) in non SURVIVORS: 0(0%)
-->>7r 19[H] 3[H] in SURVIVORS: 3(21%) in non SURVIVORS: 0(0%)
-->>8r 19[H] 6[H] in SURVIVORS: 4(28%) in non SURVIVORS: 0(0%)
-->>9r 19[H] 14[L] in SURVIVORS: 4(28%) in non SURVIVORS: 0(0%)
-->>10r 19[H] 4[L] in SURVIVORS: 3(21%) in non SURVIVORS: 0(0%)
-->>11r 19[H] 7[H] in SURVIVORS: 4(28%) in non SURVIVORS: 0(0%)
-->>12r 9[L] 18[L] 15[L] 5[L] in SURVIVORS: 4(28%) in non SURVIVORS: 0(0%)
-->>13r 20[H] 15[L] 6[H] 10[H] 3[H] in SURVIRS: 3(21%) in non SURVIVORS: 0(0%)
-->>14r 18[L] 15[L] 3[H] 5[L] in SURVIVORS: 3(21%) in non SURVIVORS: 0(0%)
```

Abbildung 5.13. Generierte Regeln (*GeneRule* log Datei). Die Beschreibung s. Abb. 5.3.

```
#Daten base name: bc_train.dat
Vector of tf: Brain_MD_ns_1 Recognized by rules for tf:1 survivors: 0 -> class tf     By rules tf: 44
Vector of tf: Brain_MD_ns_2 Recognized by rules for tf:12 survivors: 0 -> class tf    By rules tf: 4 7 8 11 15 17 19 22 25 42 43 44
Vector of tf: Brain_MD_ns_3 Recognized by rules for tf:15 survivors: 0 -> class tf    By rules tf: 4 6 8 11 15 16 18 19 21 22 24 28 41 42 43
Vector of tf: Brain_MD_ns_4 Recognized by rules for tf:12 survivors: 0 -> class tf    By rules tf: 1 11 15 16 17 18 20 21 23 27 30 46
Vector of tf: Brain_MD_ns_5 Recognized by rules for tf:9 survivors: 0 -> class tf     By rules tf: 1 4 11 15 17 18 21 24 41
Vector of tf: Brain_MD_ns_6 Recognized by rules for tf:10 survivors: 0 -> class tf    By rules tf: 1 11 12 15 18 19 24 25 35 42
Vector of tf: Brain_MD_ns_7 Recognized by rules for tf:12 survivors: 0 -> class tf    By rules tf: 1 4 7 11 12 15 19 24 25 35 38 42
Vector of tf: Brain_MD_ns_8 Recognized by rules for tf:10 survivors: 0 -> class tf    By rules tf: 1 11 15 18 20 21 23 24 46 47
Vector of tf: Brain_MD_ns_9 Recognized by rules for tf:11 survivors: 0 -> class tf    By rules tf: 4 5 7 8 14 20 22 25 35 43 45
Vector of tf: Brain_MD_ns_10 Recognized by rules for tf:9 survivors: 0 -> class tf    By rules tf: 1 11 15 18 19 21 23 24 47
Vector of tf: Brain_MD_ns_11 Recognized by rules for tf:19 survivors: 0 -> class tf   By rules tf: 9 11 13 14 16 17 20 22 24 26 27 28 30 32 33 38 39 43 45
Vector of tf: Brain_MD_ns_12 Recognized by rules for tf:11 survivors: 0 -> class tf   By rules tf: 13 14 26 27 28 30 31 32 33 39 45
Vector of tf: Brain_MD_ns_13 Recognized by rules for tf:8 survivors: 0 -> class tf    By rules tf: 1 5 6 26 28 31 32 33
Vector of tf: Brain_MD_ns_14 Recognized by rules for tf:9 survivors: 0 -> class tf    By rules tf: 9 13 27 28 29 38 39 43 45
Vector of tf: Brain_MD_ns_15 Recognized by rules for tf:0 survivors: 0                unknown !!!!!
Vector of tf: Brain_MD_ns_16 Recognized by rules for tf:2 survivors: 0 -> class tf    By rules tf: 1 41
Vector of tf: Brain_MD_ns_17 Recognized by rules for tf:13 survivors: 0 -> class tf   By rules tf: 1 4 6 7 11 15 16 17 23 24 28 33 38
Vector of tf: Brain_MD_ns_18 Recognized by rules for tf:3 survivors: 0 -> class tf    By rules tf: 1 46 47
Vector of tf: Brain_MD_ns_19 Recognized by rules for tf:14 survivors: 0 -> class tf   By rules tf: 11 15 16 20 21 23 24 27 28 37 38 39 46 47
Vector of tf: Brain_MD_ns_20 Recognized by rules for tf:8 survivors: 0 -> class tf    By rules tf: 1 15 17 18 20 23 30 46
Vector of tf: Brain_MD_ns_21 Recognized by rules for tf:4 survivors: 0 -> class tf    By rules tf: 5 7 26 37
Vector of tf: Brain_MD_ns_22 Recognized by rules for tf:2 survivors: 0 -> class tf    By rules tf: 1 47
Vector of tf: Brain_MD_ns_23 Recognized by rules for tf:6 survivors: 0 -> class tf    By rules tf: 28 29 37 38 41 42
Vector of tf: Brain_MD_ns_24 Recognized by rules for tf:8 survivors: 0 -> class tf    By rules tf: 1 11 15 18 19 21 41 42
Vector of tf: Brain_MD_ns_25 Recognized by rules for tf:18 survivors: 0 -> class tf   By rules tf: 4 5 6 7 8 9 11 12 13 16 19 22 25 29 35 42 43 44
Vector of tf: Brain_MD_ns_26 Recognized by rules for tf:18 survivors: 0 -> class tf   By rules tf: 4 5 6 7 8 11 16 17 19 21 22 24 26 28 31 32 33 37
 Recognized true 25, error 0, unk 1 of 26 vectors of tf
```

Abbildung 5.14a. Ein Teil der *DataControl* log-Datei für die neuen Regeln (Lerndatensatz) und Klasse *TF*. Die Beschreibung s. Abb. 5.4.

Vector of survivors: Brain_MD_s_1 Recognized by rules for survivors:8 tf: 0 -> class survivors	By rules survivors: 1 4 5 6 9 10 12 14
Vector of survivors: Brain_MD_s_2 Recognized by rules for survivors:0 tf: 0	unknown !!!!!
Vector of survivors: Brain_MD_s_3 Recognized by rules for survivors:0 tf: 0	unknown !!!!!
Vector of survivors: Brain_MD_s_4 Recognized by rules for survivors:4 tf: 0 -> class survivors	By rules survivors: 2 3 12 14
Vector of survivors: Brain_MD_s_5 Recognized by rules for survivors:3 tf: 0 -> class survivors	By rules survivors: 1 6 11
Vector of survivors: Brain_MD_s_6 Recognized by rules for survivors:7 tf: 0 -> class survivors	By rules survivors: 1 4 5 6 9 10 11
Vector of survivors: Brain_MD_s_7 Recognized by rules for survivors:0 tf: 0	unknown !!!!!
Vector of survivors: Brain_MD_s_8 Recognized by rules for survivors:2 tf: 0 -> class survivors	By rules survivors: 2 3
Vector of survivors: Brain_MD_s_9 Recognized by rules for survivors:0 tf: 0	unknown !!!!!
Vector of survivors: Brain_MD_s_10 Recognized by rules for survivors:9 tf: 0 -> class survivors	By rules survivors: 1 4 5 6 9 10 11 12 14
Vector of survivors: Brain_MD_s_11 Recognized by rules for survivors:7 tf: 0 -> class survivors	By rules survivors: 1 4 5 6 9 11 12
Vector of survivors: Brain_MD_s_12 Recognized by rules for survivors:0 tf: 0	unknown !!!!!
Vector of survivors: Brain_MD_s_13 Recognized by rules for survivors:2 tf: 0 -> class survivors	By rules survivors: 2 3
Vector of survivors: Brain_MD_s_14 Recognized by rules for survivors:0 tf: 0	unknown !!!!!
Recognized true 8, error 0, unk 6 of 14 vectors of survivors	

Abbildung 5.14.b Ein Teil der *DataControl* log-Datei für die neuen Regeln (Lerndatensatz) und Klasse *Survivors*. Die Beschreibung s. Abb. 5.4.

Die Löschung der Regeln hat keinen Einfluss auf die Erkennungsgüte des Lerndatensatzes.
Im nächsten Abschnitt wird die Nutzung des logischen Herangehens nach Schlüsselwort-Substitution untersucht.

5.3.3. Schlüsselwörterextraktion, Schlüsselwörter und Regelreduktion

Zuerst werden Schlüsselwörter generiert (Tabelle 5.17). Das angewandte Suchprinzip besteht darin, dass das Schlüsselwort für mindestens zwei Proteine gültig ist.

Name	SwissProt Name	REGULATION	TRANSCRIPTION REGULATION	DEFECTS	ALTERNATIVE SPLICING	PLACENTA	MODULATE	SIGNAL	GLYCOPROTEIN	CALCIUM	DISEASE MUTATION	NUCLEAR	RECEPTOR	ZINC	EC 2 / TRANSFERASE	METAL	LIVER	KIDNEY	AFFINITY	TRANSPORT	CONTAINS 1 THYROGLOBULIN TYPE-I DOMAIN	EXTRACELLULAR MATRIX	BRAIN
IMPDH2	IMD2	+																					
BAT1	BAT1			+		+			+		+						+	+	+	+			
C1NH	IC1			+				+	+		+												
hCE-2	Q9UKY3																						
COL1A2	CA21			+				+	+	+	+											+	
Diazepam inhib.	ACBP				+		+						+						+	+			
IGFBP5	IBP5							+	+				+				+	+			+		+
KIAA0275 gene	TIC2							+	+	+											+	+	+
Lim-1	LHX1											+		+		+							
MFAP4	MFA4							+	+	+												+	
PCOLCE	PCO1							+	+														
PI4-KINASE α	PI4K					+									+								+
Prostaglandin D2	P41222																						
Reelin	P78509																						
L27a	RL2A																						
hsRPB10	RPBX											+		+	+	+							
SOX2	SOX2	+	+									+											
TUBB	TBB4																						
VDAC2	POR2				+																		
Ear-3	COT1	+	+									+	+	+									
X11	APB2						+					+								+			+

Tabelle 5.17. Die gewählten Schlüsselwörter.

In Tab. 5.18 sind die gelöschten Regeln und die Merkmale, die interne Widersprüche hervorbringen, dargestellt. Die Daten, die in diesem Abschnitt verwendet werden, haben einige Besonderheiten im Ver-

gleich zu vorhergehenden Kapiteln. Die weitere Vereinfachung der Regelbasis für die Klasse "*survivors*" ist ohne Verschlechterung der Erkennungsqualität kompliziert. Deshalb werden in der Klasse *survivors* nur die "falschen" Merkmale gelöscht. Die erhaltenen Regeln werden als "endgültige" und „korrekte" Regeln bezeichnet. Alle Merkmale, die in beiden Klassen vorkommen sind nicht richtig.

#	Regel	REGULATION	TRANSCRIPTION REGULATION	DEFECTS	SIGNAL	GLYCOPROTEIN	DISEASE MUTATION	NUCLEAR	ZINC	BRAIN
3	11[L]									
13	19[L] 4[H] 15[L] 2[H]	HL	HL		H	H		HLH	H	H
16	12[L] 2[H]	H	H					HL	HL	
20	12[L] 14[H]							HL	L	H
35	20[H] 18[H]			HL	HL	HL	HL			
45	14[H] 7[L]							H		HL

Tabelle 5.18.Entfernte Regeln und Schlüsselwörter für *TF*

In Tab. 5.19 sind die Merkmale, die Widersprüche in der Klasse *survivors* (die Merkmale sind fett markiert) erzeugen, dargestellt. Diese Merkmale kann man löschen (aber keine Regeln). Die Merkmale der Klasse „*survivors*" besitzen nur eine Bedeutung. Für die Klasse *TF* werden diese Merkmale negiert. Folgende Vereinfachungen der logischen Regeln für die Klasse "TF" werden vorgenommen:

1. Die Merkmale «*ALTERNATIVE SPLICING*», «*CALCIUM*», «*EXTRACELLULAR MATRIX*" werden gelöscht
2. Die Regeln für die Klasse " *survivors* " (siehe Tabelle 5.20) werden gelöscht.
3. *DNF* – Erzeugung. (die Ergebnisse sind in der Tabelle 5.21 dargestellt)

#	ALTERNATIVE SPLICING	PLACENTA	MODULATE	CALCIUM	RECEPTOR	EC 2/ TRANSFERASE	METAL	AFFINITY	TRANSPORT	EXTRACELLULAR MATRIX
1				L						L
2	H			H	L	H	H			H
3	L		L	H	L	H	H	L	L	H
6				L						L
11		H				H				

Tabelle 5.19. Markierte Schlüsselwörter können entfernt werden. («survivors»).

#	PLACENTA	MODULATE	RECEPTOR	EC 2/ TRANSFERASE	METAL	LIVER	KIDNEY	AFFINITY	TRANSPORT	CONTAINS 1 THYRO-GLOBULIN TYPE-I DOMAIN
15r		L	L	L	L			L	L	
19r		L		L	L				L	
21r			L	L	L	L	L			L
37r			L			L	L			L
41r			L			L	L			L
42r		L							L	
44r	L	L		L					L	
47r			L			L	L			L

Tabelle 5.20. Entfernung der „TF" Regeln, die die gleichen Werte in der Klasse «survivors» haben

Regeln	***Warum ist gelöscht***	***PLACENTA***	***MODULATE***	***RECEPTOR***	***EC 2/ TRANSFE-RASE***	***METAL***	LIVER	KIDNEY	AFFINITY	***TRANSPORT***	***CONTAINS 1 THYRO-GLOBULIN TYPE-1 DOMAIN***
-->>1r 1[H]											H
-->>2r 5[H]	Leer										
-->>3r 11[L]	Leer										
-->>4r 8[H] 12[L]					L	L					
-->>5r 8[H] 9[H]	**6/9/28/29**		**H**	**H**					**H**	**H**	
-->>6r 8[H] 2[H]				H							
-->>7r 8[H] 3[L]	Leer										
-->>8r 8[H] 7[L]		L			L						
-->>9r 16[L] 2[H]				H							
-->>10r 16[L] 6[L]	Leer										
-->>11r 19[L] 12[L]					L	L					
-->>12r 19[L] 4[H] 20[H]	**6/9/28/29**			**H**			**H**	**H**			**H**
-->>14r 4[H] 9[H] 14[H]	**6/9/28/29**		**H**	**H**			**H**	**H**	**H**	**H**	**H**
-->>17r 12[L] 20[L]					L	L					
-->>18r 12[L] 3[H]					L	L					
-->>22r 12[L] 7[L]	**8/43**	**L**			**L**	**L**					
-->>23r 12[L] 15[H]					L	L					
-->>24r 12[L] 10[H]					L	L					
-->>25r 12[L] 18[H]					L	L					
-->>26r 9[H] 10[H] 18[L]	**6/9/28/29**		**H**	**H**					**H**	**H**	
-->>27r 2[H] 14[H]	**6/9/28/29**		**H**	**H**						**H**	
-->>28r 2[H] 10[H]				H							
-->>29r 2[H] 18[H]				H							
-->>30r 20[L] 18[L] 14[H]			H							H	
-->>31r 20[L] 18[L] 6[H]	Leer										
-->>32r 20[L] 18[L] 15[L]	Leer										
-->>33r 20[L] 18[L] 10[H]	Leer										
-->>34r 20[H] 15[H]	Leer										
-->>36r 20[H] 10[H] 6[L]	Leer										
-->>38r 3[L] 10[H] 6[L]	Leer										
-->>39r 3[L] 10[H] 14[H]			H							H	
-->>40r 6[L] 15[L] 10[H]	Leer										
-->>43r 6[L] 15[L] 7[L]		L			L						
-->>46r 14[H] 15[H]			H							H	

Tabelle 5.21. *DNF* - Erzeugung. Fettmarkierte Regeln sind gelöscht.

Klasse	Regeln	PLACENTA	MODULATE	RECEPTOR	EC 2/ TRANSFERASE	METAL	AFFINITY	TRANSPORT	CONTAINS 1 THYROGLOBULIN TYPE-I DOMAIN
Survivors	-->>1r 8[L] 19[H]								
	-->>2r 8[H] 2[L] 12[H] 3[H]			L	H	H			
	-->>3r 8[H] 2[L] 12[H] 9[L]		L	L	H	H	L	L	
	-->>6r 19[H] 18[L]								
	-->>11r 19[H] 7[H]	H			H				

Klasse	Regeln	PLACENTA	MODULATE	RECEPTOR	EC 2/ TRANSFERASE	METAL	AFFINITY	TRANSPORT	CONTAINS 1 THYROGLOBULIN TYPE-I DOMAIN
TF	-->>1r 1[H]								H
	-->>4r 8[H] 12[L]				L	L			
	-->>6r 8[H] 2[H]			H					
	-->>8r 8[H] 7[L]	L			L				
	-->>9r 16[L] 2[H]			H					
	-->>11r 19[L] 12[L]				L	L			
	-->>17r 12[L] 20[L]				L	L			
	-->>18r 12[L] 3[H]				L	L			
	-->>23r 12[L] 15[H]				L	L			
	-->>24r 12[L] 10[H]				L	L			
	-->>25r 12[L] 18[H]				L	L			
	-->>28r 2[H] 10[H]			H					
	-->>29r 2[H] 18[H]			H					
	-->>30r 20[L] 18[L] 14[H]		H					H	
	-->>39r 3[L] 10[H] 14[H]		H					H	
	-->>43r 6[L] 15[L] 7[L]	L			L				
	-->>46r 14[H] 15[H]		H					H	

Tabelle 5.22. Finalversion der Regeln für Hirntumor Datensatz

Nach der Vereinfachung der Regelbasis erhält man 5 Regeln für die Klasse "*survivors*" und 17 Regeln für die Klasse *TF*. Eine Zusammenfassung der Ergebnisse ist in der Tabelle 5.22 dargestellt. In Abb. 5.15 und 5.16 sind die Ergebnisse der Erkennung dargestellt. Auf der Basis von 22 logischen Regeln können 31 von 40 Objekten des Lerndatensatzes und 14 von 20 Objekten des Kontrolldatensatzes richtig unterschieden werden. Insgesamt sind 45 von 60 Objekten richtig erkannt wurden.

```
#Knowledge base name: part4
#Daten base name: bc_test.dat
Vector of tf: Brain_MD_ns_27 Recognized by rules for tf:1 survivors: 0 -> class tf          By rules tf: 1
Vector of tf: Brain_MD_ns_28 Recognized by rules for tf:0 survivors: 2 -> class survivors    By rules survivors: 1 6                ERROR !!!!
Vector of tf: Brain_MD_ns_29 Recognized by rules for tf:2 survivors: 0 -> class tf          By rules tf: 1 46
Vector of tf: Brain_MD_ns_30 Recognized by rules for tf:8 survivors: 0 -> class tf          By rules tf: 1 4 6 11 17 18 24 28
Vector of tf: Brain_MD_ns_31 Recognized by rules for tf:6 survivors: 0 -> class tf          By rules tf: 11 17 24 25 39 43
Vector of tf: Brain_MD_ns_32 Recognized by rules for tf:3 survivors: 0 -> class tf          By rules tf: 11 18 24
Vector of tf: Brain_MD_ns_33 Recognized by rules for tf:7 survivors: 0 -> class tf          By rules tf: 4 11 17 18 23 24 25
Vector of tf: Brain_MD_ns_34 Recognized by rules for tf:3 survivors: 0 -> class tf          By rules tf: 6 9 28
Vector of tf: Brain_MD_ns_35 Recognized by rules for tf:2 survivors: 2 -> class survivors   By rules tf: 18 24      By rules survivors: 1 6
                                                                                                                    ERROR !!!!
Vector of tf: Brain_MD_ns_36 Recognized by rules for tf:6 survivors: 0 -> class tf          By rules tf: 9 11 17 18 25 29
Vector of tf: Brain_MD_ns_37 Recognized by rules for tf:1 survivors: 0 -> class tf          By rules tf: 6
Vector of tf: Brain_MD_ns_38 Recognized by rules for tf:4 survivors: 0 -> class tf          By rules tf: 11 18 24 28
Vector of tf: Brain_MD_ns_39 Recognized by rules for tf:6 survivors: 0 -> class tf          By rules tf: 11 23 24 25 28 29
 Recognized true 11, error 2, unk 0 of 13 vectors of tf
Vector of survivors: Brain_MD_s_15 Recognized by rules for survivors:0 tf: 11 -> class tf   By rules tf: 1 4 6 8 9 11 17 18 23 24 28   ERROR !!!!
Vector of survivors: Brain_MD_s_16 Recognized by rules for survivors:0 tf: 5 -> class tf    By rules tf: 9 11 17 18 30              ERROR !!!!
Vector of survivors: Brain_MD_s_17 Recognized by rules for survivors:2 tf: 0 -> class survivors                By rules survivors: 2 3
Vector of survivors: Brain_MD_s_18 Recognized by rules for survivors:2 tf: 5 -> class survivors   By rules tf: 4 6 18 24 28By rules survivors: 6 11
Vector of survivors: Brain_MD_s_19 Recognized by rules for survivors:2 tf: 4 -> class survivors   By rules tf: 4 17 18 30  By rules survivors: 6 11
Vector of survivors: Brain_MD_s_20 Recognized by rules for survivors:0 tf: 2 -> class tf   By rules tf: 1 46                      ERROR !!!!
Vector of survivors: Brain_MD_s_21 Recognized by rules for survivors:2 tf: 8 -> class tf   By rules tf: 1 4 6 17 24 28 30 39By rules survivors: 6 11
                                                                                                                    ERROR !!!!
 Recognized true 3, error 4, unk 0 of 7 vectors of survivors
```

Abbildung 5.15. Eine Log-Datei von *DataControl* für reduzierte Regeln und Kontrolledatensatz. Erklärung s. Abb. 5.4.

```
#Knowledge base name: part4
#Daten base name: bc_train.dat
Vector of tf: Brain_MD_ns_1 Recognized by rules for tf:0 survivors: 0                                                    unknown !!!!!
Vector of tf: Brain_MD_ns_2 Recognized by rules for tf:6 survivors: 0 -> class tf        By rules tf: 4 8 11 17 25 43
Vector of tf: Brain_MD_ns_3 Recognized by rules for tf:8 survivors: 0 -> class tf        By rules tf: 4 6 8 11 18 24 28 43
Vector of tf: Brain_MD_ns_4 Recognized by rules for tf:7 survivors: 0 -> class tf        By rules tf: 1 11 17 18 23 30 46
Vector of tf: Brain_MD_ns_5 Recognized by rules for tf:6 survivors: 0 -> class tf        By rules tf: 1 4 11 17 18 24
Vector of tf: Brain_MD_ns_6 Recognized by rules for tf:5 survivors: 0 -> class tf        By rules tf: 1 11 18 24 25
Vector of tf: Brain_MD_ns_7 Recognized by rules for tf:5 survivors: 0 -> class tf        By rules tf: 1 4 11 24 25
Vector of tf: Brain_MD_ns_8 Recognized by rules for tf:6 survivors: 0 -> class tf        By rules tf: 1 11 18 23 24 46
Vector of tf: Brain_MD_ns_9 Recognized by rules for tf:4 survivors: 0 -> class tf        By rules tf: 4 8 25 43
Vector of tf: Brain_MD_ns_10 Recognized by rules for tf:5 survivors: 0 -> class tf       By rules tf: 1 11 18 23 24
Vector of tf: Brain_MD_ns_11 Recognized by rules for tf:8 survivors: 0 -> class tf       By rules tf: 9 11 17 24 28 30 39 43
Vector of tf: Brain_MD_ns_12 Recognized by rules for tf:3 survivors: 0 -> class tf       By rules tf: 28 30 39
Vector of tf: Brain_MD_ns_13 Recognized by rules for tf:3 survivors: 0 -> class tf       By rules tf: 1 6 28
Vector of tf: Brain_MD_ns_14 Recognized by rules for tf:5 survivors: 0 -> class tf       By rules tf: 9 28 29 39 43
Vector of tf: Brain_MD_ns_15 Recognized by rules for tf:0 survivors: 0                                                   unknown !!!!!
Vector of tf: Brain_MD_ns_16 Recognized by rules for tf:1 survivors: 0 -> class tf       By rules tf: 1
Vector of tf: Brain_MD_ns_17 Recognized by rules for tf:8 survivors: 0 -> class tf       By rules tf: 1 4 6 11 17 23 24 28
Vector of tf: Brain_MD_ns_18 Recognized by rules for tf:2 survivors: 0 -> class tf       By rules tf: 1 46
Vector of tf: Brain_MD_ns_19 Recognized by rules for tf:6 survivors: 0 -> class tf       By rules tf: 11 23 24 28 39 46
Vector of tf: Brain_MD_ns_20 Recognized by rules for tf:6 survivors: 0 -> class tf       By rules tf: 1 17 18 23 30 46
Vector of tf: Brain_MD_ns_21 Recognized by rules for tf:0 survivors: 0                                                   unknown !!!!!
Vector of tf: Brain_MD_ns_22 Recognized by rules for tf:1 survivors: 0 -> class tf       By rules tf: 1
Vector of tf: Brain_MD_ns_23 Recognized by rules for tf:2 survivors: 0 -> class tf       By rules tf: 28 29
Vector of tf: Brain_MD_ns_24 Recognized by rules for tf:3 survivors: 0 -> class tf       By rules tf: 1 11 18
Vector of tf: Brain_MD_ns_25 Recognized by rules for tf:8 survivors: 0 -> class tf       By rules tf: 4 6 8 9 11 25 29 43
Vector of tf: Brain_MD_ns_26 Recognized by rules for tf:7 survivors: 0 -> class tf       By rules tf: 4 6 8 11 17 24 28
 Recognized true 23, error 0, unk 3 of 26 vectors of tf
Vector of survivors: Brain_MD_s_1 Recognized by rules for survivors:2 tf: 0 -> class survivors   By rules survivors: 1 6
Vector of survivors: Brain_MD_s_2 Recognized by rules for survivors:0 tf: 0                                              unknown !!!!!
Vector of survivors: Brain_MD_s_3 Recognized by rules for survivors:0 tf: 0                                              unknown !!!!!
Vector of survivors: Brain_MD_s_4 Recognized by rules for survivors:2 tf: 0 -> class survivors   By rules survivors: 2 3
Vector of survivors: Brain_MD_s_5 Recognized by rules for survivors:3 tf: 0 -> class survivors   By rules survivors: 1 6 11
Vector of survivors: Brain_MD_s_6 Recognized by rules for survivors:3 tf: 0 -> class survivors   By rules survivors: 1 6 11
Vector of survivors: Brain_MD_s_7 Recognized by rules for survivors:0 tf: 0                                              unknown !!!!!
Vector of survivors: Brain_MD_s_8 Recognized by rules for survivors:2 tf: 0 -> class survivors   By rules survivors: 2 3
Vector of survivors: Brain_MD_s_9 Recognized by rules for survivors:0 tf: 0                                              unknown !!!!!
Vector of survivors: Brain_MD_s_10 Recognized by rules for survivors:3 tf: 0 -> class survivors  By rules survivors: 1 6 11
Vector of survivors: Brain_MD_s_11 Recognized by rules for survivors:3 tf: 0 -> class survivors  By rules survivors: 1 6 11
Vector of survivors: Brain_MD_s_12 Recognized by rules for survivors:0 tf: 0                                             unknown !!!!!
Vector of survivors: Brain_MD_s_13 Recognized by rules for survivors:2 tf: 0 -> class survivors  By rules survivors: 2 3
Vector of survivors: Brain_MD_s_14 Recognized by rules for survivors:0 tf: 0                                             unknown !!!!!
 Recognized true 8, error 0, unk 6 of 14 vectors of survivors
```

Abbildung 5.16. Log Datei für reduzierte Regeln und Lerndatensatz. Erklärung s. Abb. 5.4.

5.4. Vergleichsanalyse der Ergebnissen

In der Tabelle 5.23 sind die Ergebnisse für die Leukämie-Daten dargestellt.

Methode		Genanzahl	Fehler/Ohne Lösung
Logische Regeln	*GeneRule*	10	1
Logische Regeln + Schlüsselwörterbearbeitung		10	0
Diskretisierung und Geneselektion Methode	[Li2002]	1	3
SVM	[Furey2000]	25-1000	2-4
Diskretisierung und diskrimination Methode	[Nguyen2002]	813	0
		94	1
weighting/voting	[Bijani2003]	16	0
Weighted/voting	[Golub1999]	50	5

Tabelle 5.23. Vergleichsanalyse der Methodenergebnisse für „Leukämie“.

Ohne Fehler arbeiten drei Methoden. Unter den drei fehlerfreien Methoden befindet sich auch *GeneRule*. Der Algorithmus von *GeneRule* ist der Beste, da er das fehlerfreie Ergebnis mit der geringsten Anzahl von Genen liefert.

Die Ergebnisse für den Darmkrebs sind in der Tabelle 5.24 dargestellt. Es ist offensichtlich, dass *GeneRule* die besten Ergebnisse liefert, da es die kleinste Anzahl von Genen verwendet.

Methode		Genesanzahl	Fehler /Ohne Lösung
Logische Regeln	*GeneRule*	11	4
Logische Regeln + Schlüsselwörterbearbeitung		11	3
Diskretisierung und Geneselektion Methode	[Li2002]	35	5
SVM	[Furey2000]	1000	6

Tabelle 5.24. Vergleichsanalyse der Methodenergebnisse für „Darmkrebs".

Die Ergebnisse für Gehirn-Krebs sind in der Tabelle 5.25 dargestellt. Die Ergebnisse für *GeneRule* verschlechterten sich nach der Durchführung der logischen Analyse mittels Schlüsselwortsubstitution. Die Ergebnisse für *GeneRule* waren bis zur Durchführung der logischen Analyse vergleichbar mit den Ergebnissen der besten Algorithmen.

Methode		Fehler/Ohne Lösung
Logische Regeln	*GeneRule*	12
Logische Regeln + Schlüsselwörterbearbeitung		15
TrkC	[Pomeroy2002]	20
Staging		19
SVM		15
SPLASH		15
Weighted Voting		14
k-NN		13
Combined model I Staging, *k-NN* and TrkC		12
Combined model II *SVM*, *k-NN* and TrkC		12

Tabelle 5.25. Vergleichsanalyse der Methodenergebnisse für „Hirntumor"

Nach der Durchführung der logischen Analyse sind die Ergebnisse schlechter als für *k-NN* und Weighted Voting geworden aber identisch mit *SVM* und *SPLASH*. Die erhaltenen Ergebnisse sind mit den Unexaktheiten in den Daten verbunden. Die Autoren [Pomeroy2002] zeigen, dass es Fehlermöglichkeiten bei der Verifizierung gab. Für die Verbesserung von Λ wird die nochmalige Verifizierung gefordert. Bei der Verifizierung kann man die vorliegenden Ergebnisse der Objektklassifikation nutzen.

5.5. Analysis der Genlokalisierung

Die Analyse der Ergebnisse kann man in verschiedenen Richtungen interpretieren. Ein Versuch, die selektierten Gene mit der Topologie oder den bekannten genetischen Defekten zu verbinden, war erfolglos (Tabelle 5.26).

Gene	Lokalisierung		Wo
Macmarcks	1p34.3	t(1;19) translocation in B-lineage ALL	[Troussard1995]
		JT (Jumping translocations) in which the long arm of chromosome 1 distal to q21 is translocated to the terminal region of chromosome 10	[Okita2000]
MCL1	1q21. 2	t(1;19) translocation in B-lineage ALL	[Troussard1995]
		JT (Jumping translocations) in which the long arm of chromosome 1 distal to q21 is translocated to the terminal region of chromosome 10	[Okita2000]
Cystatin C	20p11.2	-	-
ATP6C	16p13.3	a case of AML with t(16;21)	[Okita2000]
Cyclin D3	6p21	Almost all of the genes lie in regions that have been identified previously to contain abnormalities in AML or other forms of leukemia. Furthermore, three of the genes are encoded within 5q11–31, four are in the 2q region, two are within 1q32–26, and two others are found at 6p.	[Thomas2001]
myb	6p22		
Azurocidin	19p13.3	t(1;19) translocation in B-lineage ALL	[Troussard1995]
p62	5q31	Deletions of chromosome 5q (5q-) in AML.	[Sun1991]
		Of note, the region 5q11–31 is frequently lost in AML and known to influence prognosis (Shipley et al. 1996; El-Rifai et al. 1997; Van den Berghe and Michaux 1997).	[Thomas2001]
IOTA	-	-	-
TOP2B	3p24	t(3;21)	[Zeng1997]

Tabelle 5.26. Selektierte Gene (Leukämie Daten) und genetische Defekte

5.6. Praktische Genanalysis

Die Information über die ausgewählte Gene wurden in den biologischen Datenbanken gesucht. Die Ergebnisse für die Leukämie sind in der Tabelle 5.27 dargestellt. Aus der Tabelle kann man sehen, dass die volle Information für die Gene nur in den Datenbanken *Swiss-Prot/TrEMBL* gefunden wurde.

Desweiteren wurde versucht die in [Knudsen2001] beschriebene Methodik für die Analyse der leukämischen Daten auszunutzen. Die Ergebnisse sind in Abb.5.17 dargestellt. Es ist aufgrund der Datenlage nicht möglich, Aussagen über die Wechselwirkung der gefundenen Gene zu machen. Die Gründe dafür sind: die Unvollständigkeit der biologischen Datenbanken und die Methodik selbst. Eines der Hauptprinzipien der Durchführung der Analyse ist die bedeutende Reduktion der Anzahl von gefundenen Gene - die Entdeckung der spezifischen Marker. Bei der bedeutenden Verengung des Gebietes der untersuchten Gene bekommen wir die Gene, die für verschiedene Prozesse verantwortlich sind. Die erhaltenen Ergebnisse sind interessant für die Spezialisten des entsprechenden Sachgebietes. Die Ergebnisse sind aber nicht geeignet, um Wechselwirkungen zwischen den Genen aufzudecken

Gene	Swiss-Prot/TrEMBL	Transpath	KEGG
TP6C	VATL	-	K02148
Azurocidin	CAP7	-	566
Cyclin D3	CGD3	MO0000031198	896
C-myb	MYB	MO000009619	-
Cystatin C	CYTC	-	1471
MCL1	MCL1	-	4170
Macmarcks	MRP	-	65108
p62	Q13501	MO0000022279	51164
IOTA	PSA6	-	-
TOP2B	TP2B	-	K03164
Total	10	3	8

Tabelle 5.27. Selektierte Gene in biologischen Datenbanken (Datensatz „Leukämie“)

Desweiteren wurde versucht auch bekannten und in ‚transpath‘ gespeicherten Trends der Aktivierung oder Deaktivierung zu erkennen. Die erhaltenen Ergebnisse sind nicht positiv, da nur für 3 Gene ein Schema der (De-) Aktivierung in ‚transpath‘ gefunden wurde. Bei der kleinen Entfernung (*n = 5*) zwischen den Proteinen sind die erhaltenen Ergebnisse in der Abbildung 5.17 dargestellt sind. Im Vergleich mit den praktischen Ergebnissen (5.18 im Text) erhält man widersprechende Ergebnisse. Ergebnisse, die aus Mikroarray bezogenen Ergebnissen entsprechen, erhält man bei der Anwendung von größeren Entfernungen zwischen den Proteinen. Das dabei erhaltene Schema der Wechselwirkung der Proteine ist außerordentlich kompliziert. Die Methodik für die Analyse des komplizierten Wechselwirkungsschemas und ihre Anwendung für die Analyse der praktischen Ergebnisse ist perspektivisch. Die Erarbeitung dieses Algorithmus ist aus zeitlichen Gründen hinter den Rahmen der Arbeit geblieben.

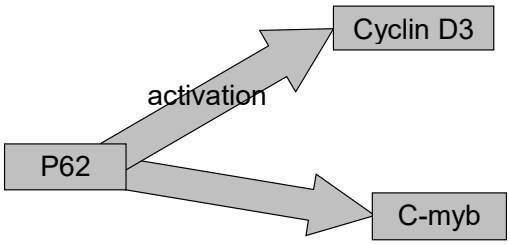

Abbildung 5.18. Analysis der Zusammenwirkung *(n = 5)*

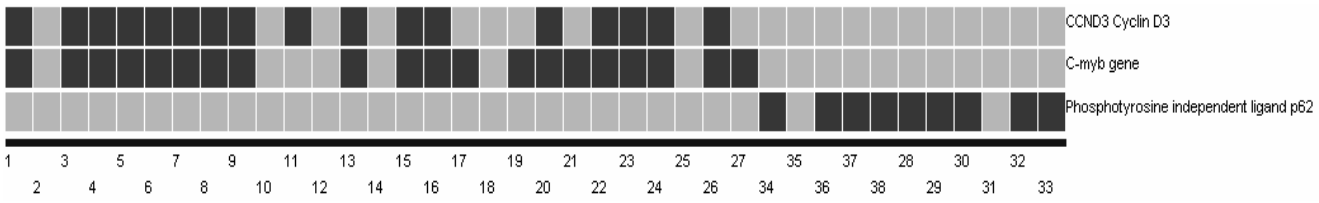

Abbildung 5.19. Genexpression in Mikroarrays (Datensatz „Leukämie“). Der p62 Wert ist hoch und die Werte der anderen Gene sind niedrig.

5.7. Fazit

1. Die logische Regeln ermöglichen eine doppelte Datenanalyse: die Analyse der Merkmale und der Objekte.

2. Durch die entwickelte Methodik der logischen, auf Schlüsselwörtern basierenden Regelanalyse sind leicht interpretierbare Ergebnisse zu erhalten.

3. Die entwickelte Methodik der Analyse der Genwechselwirkungen durch *Transpath*-Reaktionen war in der gegebenen Zeit nicht möglich. Teilweise ist dies bedingt durch die Unvollkommenheit der existierenden biologischen Wissensbasen.

4. Die Nutzung einiger publizierter Methodiken (z.B. Geneslokalisierung und [Knudsen2002]) ermöglicht es, interpretierte Ergebnisse zu erhalten, aber die automatisierte Analyse dieser Ergebnisse ist nicht möglich.

5. Die neu entwickelte Methodik wurde für die Analyse von drei aus der Literatur bekannten Datensätzen verwendet. Der erste Datensatz „Leukämie“ wurde am häufigsten zitiert und für vergleichende Analysen verwendet. Die Anwendung der Methodik der Diskretisierung und der Auswahl der Merkmale, der Erzeugung und der Validation der Regeln hat sich als wirksam erwiesen für die Lösung der Klassifikationsaufgabe bei allen drei Datensätzen.

6. Die Regelvalidationsmethodik hat in zwei Fällen zur Verbesserung der Ergebnisse geführt (Leukämie und Darmkrebs). Die dabei erhaltenen Ergebnisse sind die Besten unter den bisher veröffentlichten .

6 Zusammenfassung

- cDNA Arrays und high-density oligonucleotide Chips sind neuartige Biotechniken, die zunehmend benutzt werden. Da die Arraytechnologie die Expressionsmessung von Tausenden Genen gleichzeitig ermöglicht, wird erwartet, dass damit ein bedeutender Beitrag zum Fortschritt in der biologischen und medizinischen Grundlagenforschung möglich ist. Der Mangel der adäquaten statistischen Methoden für die Arraydatenanalyse bleibt ein Hindernis für die Erschließung des Potenzials dieser viel versprechenden Methoden. Die Modifikation der existierenden statistischen Methoden oder die Entwicklung der neuen Techniken ist notwendig für eine leistungsfähige Arraydatenanalyse.

- Logische Algorithmen sind potenziell fähig, alle drei Probleme, die bei der Analyse von Arraydaten entstehen, zu lösen:
  - Sie sind fähig, mit nahezu beliebigen Datenformaten zu arbeiten.
  - Sie sind gegen zufällige Messfehler (Rauschen) weniger empfindlich als andere Algorithmen.
  - Sie geben klar verständliche, interpretierbare Ergebnisse.

- Die existierenden logischen Algorithmen haben eine Reihe von Mängeln. Der Algorithmus „*Kora*" macht keine volle Analyse aller relevanten Kombinationen von Merkmalen. Die Einführung der bekannten zusätzlichen Beschränkungen (z.B. des Konjunktionsranges) kann die Ergebnisse wesentlich verschlechtern. Die populären Baumkonstruktionsalgorithmen verwenden einfache Algorithmen, die nicht immer zu genauen Ergebnissen führen. Diese Algorithmen konstruieren nur einen Baum, der nicht immer der beste ist. Im Falle großer Merkmalanzahl kann die Konstruktion des optimalen Baumes praktisch unmöglich sein. Die Auswahl nur eines einzigen Baumes ist eine Schwäche der bekannten Baumkonstruktionsalgorithmen. Diese muss vermieden werden. Der Versuch, das Problem durch die Aggregation zu lösen, führt zum Verlust der Durchsichtigkeit und Verständlichkeit der Ergebnisse. Ein Schlüssel der Genauigkeitsverbesserung ist die mögliche Instabilität der Vorhersagemethode, d.h. kleine Veränderungen im Lerndatensatz führen zu großen Veränderungen im Klassifikator. Die Entwicklung einer neuen Technik der direkten Regelerzeugung war notwendig.

- Es gibt vier statistische Problemen:
  1. Identifizierung der "Markergene";
  2. Klassifikation von Objekten in bekannte Klassen;
  3. Identifizierung von neuen, noch unbekannten Klassen;
  4. Analyse und Validierung des regelbasierten Wissens durch existierende Wissensbasen.

  Ausgehend von den o.g. Problemen wurde eine neue Regelextraktionsmethode entwickelt mit folgenden Komponenten:
  - Diskretisierung und der Selektion der Merkmale;
  - Extraktion von logischen Regeln
  - Validation der logischen Regeln

- Da die traditionellen Methodiken der Datendiskretisierung eine geringe Effektivität zeigen, wurde für die Diskretisierung der Ausgangsdaten eine wirksamere statistische Technik verwendet. Damit ist eine Datennormalisierung überflüssig.

- Für den neuen Regelsuchalgorithmus wurden zwei bekannte Methoden, nämlich die Entscheidungsbaumkonstruktion und „*Kora*“, kombiniert. Der entwickelte Algorithmus verwendet die folgenden Parameter:
    - die Klasse, für die die Suche durchgeführt wird;
    - die minimale Anzahl der Objekte, die durch die Regel richtig klassifiziert werden;
    - die maximale Anzahl der Objekte, die durch die Regel falsch klassifiziert werden;
    - ein Bewertungsmaß für Konjunktion;
    - eine Methode der Merkmalssortierung.

  Für die Realisierung des gegebenen Herangehens war auch die Erarbeitung einer Schlussfolgerungsmethodik erforderlich. Hierfür wurde die Abstimmung (*voting*) gewählt

- Für die Reduktion und Analyse der logischen Regeln sind drei Techniken entwickelt, implementiert und genutzt worden:
    - Nutzung der Regeln der diskreten Mathematik, Algebra und Logik: Erzeugung der disjunktiven Normalform (*DNF*)
    - Clusteranalyse im Raum der Regeln;
    - Logische Analyse der Regeln nach Substitution der Merkmale (Gene) durch ihre Beschreibungen (Schlüsselwörter).

- Die entwickelten Methoden wurden in modernen Programmtechnologien (*Java 2,XML, SQL*) realisiert. Für die Analyse und Validierung der erhaltenen Regeln durch Extraktion von Schlüsselwörtern und Netzwerkkonstruktion wurden bekannte Datenbanken (*Swiss-Prot/TrEMBL*, *Transpath* und *KEGG*) verwendet.
  Die neu entwickelte Methodik wurde anhand von 3 Datensätzen geprüft. Der Datensatz „Leukämie“ wurde genutzt, weil er am häufigsten zitiert wird, die zwei anderen Datensätze (Darm- und Hirn-Tumor), weil bisher abgewandte Klassifikationsalgorihmen zu unbefriedigenden Ergebnissen führten. Die neu entwickelten Methoden erwiesen sich für alle drei Datensätze als leistungsfähig. In einigen Fällen waren die mit der neuen Methode erhaltenen Ergebnisse die Besten unter den bisher publizierten Ergebnissen.

  Die entwickelte Methodik der logischen Regelanalyse erzeugt Ergebnisse, die leicht vom Fachexperten interpretiert werden können.

7 Literaturverzeichnis

[Aigner1996] M.Aigner: «*Diskrete Mathematik*» (1996) Vieweg, Wiesbaden.

[Albert2003] S. Albert, S. Gaudan, H. Knigge, A. Rätsch, A. Delgado, B. Huhse, H. Kirsch, M. Albers, D. Rebholz-Schuhmann and M. Kögl: «*Computer-assisted generation of a protein-interaction database for nuclear receptors*» (2003 Aug) **Mol Endocrinol**, Band 17, 8, Seiten 1555-1567

[Alizadeh2000] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Losses, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. Hudson Jr, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, and L. M. Staudt: «*Different types of diffuse large b-cell lymphoma identified by gene expression profiling*» (2000) **Nature**, 403, Seiten 503-511

[Alon1999] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack and A.J. Levine : «*Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays*» (1999) **Proc Natl Acad Sci U S A**, 96, Seiten 6745-6750

[Aronson2001] A.R. Aronson: «*Effective mapping of biomedical text to the UMLS Metathesaurus an in vitro study relevant to bone marrow purging*» (1991 Mar 15) **Proc Natl Acad Sci U S A**, Band 88, 6, Seiten 2351-2355

[Bellamy1997] J.E. Bellamy: «*Fuzzy systems approach to diagnosis in the postpartum cow*» (1997 Feb 1) **J Am Vet Med Assoc**, Band 210, 3, Seiten 397-401

[Ben-Dor1999] A.Ben-Dor, R.Shamir, Z. Yakhini: «*Clustering gene expression patterns*» (1999) **J Comput Biol**, 6, Seiten 281-297

[Ben-Dor2000] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, Z. Yakhini: «*Tissue classification with gene expression profiles*» (2000) **J Comput Biol**, 7, Seiten 559-583

[Berrar2003] D.P. Berrar, C.S. Downes, W. Dubitzky: «*Multiclass cancer classification using gene expression profiling and probabilistic neural networks*» (2003) **Pac Symp Biocomput**, Seiten 5-16

[Bijlani2003] R. Bijlani, Y. Cheng, D.A. Pearce, A.I. Brooks, M. Ogihara: «*Prediction of biologically significant components from microarray data Independently Consistent Expression Discriminator (ICED)*» (2003 Jan) **Bioinformatics**, Band 19, 1, Seiten 62-70

[Bloch1995] A. Bloch, X.M. Liu, L.G. Wang: «*Regulation of c-myb expression in ML-1 human myeloblastic leukemia cells by c-ets-1 protein*» (1995) **Adv Enzyme Regul**, Band 35, Seiten 35-41

[Bongard1967] M.M.Bongard: «*Problema uznavania (in russ.)*» (1967), Nauka, Moscow,

[Brazma2001] A. Brazma, P. Hingamp, J. Quackenbush, G. Sherlock, P. Spellman, C. Stöckert, J. Aach, W. Ansorge, C.A.Ball, H.C. Causton et al. : «*Minimum information about a microarray experiment (MIAME)—toward standards for microarray data*» (2001) **Nature Genetics**, 29, Seiten 65–371

[Breiman1984] L. Breiman, J. H. Friedman, R. Olshen, and C. J. Stone: «*Classification and regression trees*» (1984), The Wadsworth statistics/probability series. Wadsworth International Group

[Breiman1996] L. Breiman: «*Bagging predictors*» (1996) **Machine Learning**, 24, Seiten 23-140

[Brown1994] M.A. Brown, M. Harrison-Smith, E. DeLuca, C.G. Begley, N.M. Gough: «*No evidence for GM-CSF receptor alpha chain gene mutation in AML-M2 leukemias which have lost a sex chromosome*» (1994 Oct) **Leukemia**, Band 8, 10, Seiten 1774-1779

[Brown2000] M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. Ares Jr., and D. Haussler: «*Knowledge-based analysis of microarray gene expression data by using suport vector machines*» (2000) **Proc Natl Acad Sci U S A**, 97, Seiten 62-267

[Burges1998] C.Burges: «*A tutorial on support vector machines for pattern recognition*» (1998) **Data Mining and Knowledge Discovery**, 2, Seiten 21-167

[Calabretta1991] B. Calabretta, R.B. Sims, M. Valtieri, D. Caracciolo, C. Szczylik, D. Venturelli, M. Ratajczak, M. Beran, A.M. Gewirtz: «*Normal and leukemic hematopoietic cells manifest differential sensitivity*

to inhibitory effects of c-myb antisense oligodeoxynucleotides an in vitro study relevant to bone marrow purging» (1991 Mar 15) **Proc Natl Acad Sci U S A**, Band 88, 6, Seiten 2351-2355

[Carp1976] V.P.Carp, P.B.Kunin: «*Method of the directed learning in method of M.M.Bongard in oncologic diagnostic (in russ.)*» (1976), in **Applied problems of learning**, Moscow, Seiten 7-13

[Chiang2003] J.H. Chiang, H.C. Yu: «*MeKE: discovering the functions of gene products from biomedical literature via sentence alignment*» (2003 Jul 22) **Bioinformatics**, Band 19, 11, Seiten 1417-1422

[Chiang2004] J.H. Chiang, H.C. Yu, H.J. Hsu: «*GIS: a biomedical text-mining system for gene information discovery*» (2004 Jan 1) **Bioinformatics**, Band 20, 1, Seiten 120-121

[Chow2001] M.L. Chow, E.J. Moler, I.S. Mian: «*Identifying marker genes in transcription profiling data using a mixture of feature relevance experts*» (2001 Mar 8) **Physiol Genomics**, Band 5, 2, Seiten 99-111

[Clare2002] A.Clare and R. D. King: «*Machine learning of functional class from phenotype data*» (2002) **Bioinformatics**, Band 18, 1, Seiten 160-166

[Cover1967] T.M.Cover and P.E.Hart: «*Nearest neighbor pattern classification*» (1967) IEEE TransInformTheor., 13, Seiten 1-27

[de Bruijn2002] B. de Bruijn, J. Martin: «*Getting to the (c)ore of knowledge mining biomedical literature*» (2002 Dec 4) **Int J Med Inf**, Band 67, 1-3, Seiten 7-18

[DeRisi1997] J.L. DeRisi, V.R. Iyer and RO Brown: «*Exploring the metabolic and genetic control of gene expression on a genomic scale*» (1997) **Science**, 278, Seiten 680-686

[Dettling2002] M. Dettling and P. Biihlmann: «*Supervised clustering of genes*» (2002) **Genome Biology**, Band3, 12

[Djuck2001] V.Djuck, A. Samoilenko: «*Data Mining (in russ.)*» (2001), Piter, St.Petersburg,

[Dougherty1995] J.Dougherty, R.Kohavi and M.Sahami: «*Supervised and unsupervised discretization of continuous features*» (1995), Proceedings of the Twelfth International Conference on Machine Learning Morgan Kaufmann, San Francisco CA, Seiten 94—202

[Dudoit2000] S.Dudoit, J. Fridlyand and T.P. Speed: «*Comparison of discrimination methods for the classification of tumors using gene expression data*» (2000) Technical report # 576, Department of Statistics University of California, Berkeley

[Eisen1998] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein: «*Cluster analysis and display of genome-wide expression patterns*» (1998) **Proc Natl Acad Sci U S A**, 95, Seiten 14863-14868

[Erkeland2003] S. J Erkeland, M. Valkhof, C. Heijmans-Antonissen, R. Delwel, P. J M Valk, M. H A Hermans, I. P Touw: «*The gene encoding the transcriptional regulator Yin Yang 1 (YY1) is a myeloid transforming gene interfering with neutrophilic differentiation*» (2003 Feb 1) **Blood**, Band 101, 3, Seiten 1111-1117

[Fayyad1993] U.Fayyad and K. Irani : «*Multi-interval discretization of continuous-valued attributes for classification learning*» (1993), Proceedings of the 13th International Joint Conference on Artificial Intel-ligenceMorgan Kaufmann, San Francisco CA, Seiten 1022-1029

[Fisher1936] R. A. Fisher.: «*The use of multiple measurements in taxonomic problems*» (1936) **Annal of Eugenics**, 7, Seiten 79-188

[Fix1951] E. Fix and J. Hodges. : «*Discriminatory analysis, nonparametric discrimination consistency properties*» (1951) Technical report, Randolph Field Texas USAF School of Aviation Medicine

[Furey2000] T.S.Furey, N.Cristianini, N.Duffy, D.W.Bednarski, M. Schummer and D.Haussler : «*Support vector machine classi-fication and validation of cancer tissue samples using microarray expression data*» (2001) **Bioinformatics**, 16, Seiten 906-914

[Golub1999] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M.A. Caligiuri, C.D. Bloomfield, and E. S. Lander. : «*Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring*» (1999 October 15) **Science**, 286, Seiten 31-537

[Griffiths1996] A. J. F. Griffiths, J. H. Miller, D. T. Suzuki, R. C. Lewontin, and W. M. Gelbart: «*An Introduction to Genetic Analysis*» (1996), 6th edition, WHFreeman and Company, New York

[Guthke2002] R. Guthke, W. Schmidt-Heck, D. Hahn, M.Pfaff: «*Gene expression data mining for functional genomics using fuzzy technology*» (2002), in International Series in **Intelligent Technologies**

Advances in Computational Intelligence and Learning Methods and Applications, Boston Kluwer Academic Publishers, Seiten 475-487

[Guthke1998] R. Guthke, W. Schmidt-Heck, M. Pfaff: "*Knowledge acquisition and knowledge based bioprocess control*" (1998). **J. of Biotechnol**. 65, Seiten 37-46.

[Han2000] J. Han and M. Kamber: «*Data Mining: Concepts and Techniques*» (2000), Chapter 7 Classification and Prediction.

[Herzel2001] H. Herzel, D. Beule, S. Kielbasa, J. Korbel, C. Sers, A. Malik, H. Eickhoff, H. Lehrach, J. Schuchhardt: «*Extracting information from cDNA arrays*» (2001) **Chaos**, 11 (1), Seiten 8–107

[Horstmann2003a]. C.Horstamann und G. Cornell: «*Core Java 2*» (2003) Band 1-Fundamentals, Sun Microsystems Press, A Prentice Hall Title.

[Horstmann2003b]. C.Horstamann und G. Cornell: «*Core Java 2*» (2003) Band 2- Advanced Features, Sun Microsystems Press, A Prentice Hall Title.

[Huber2002] W. Huber, A. Heydebreck, A. Poustka and M. Vingron: «*Variance stabilization applied to microarray data calibration and to the quantification of differential expression*» (2002) **Bioinformatics**, Band 18, Suppl. 1, Seiten 96–104

[Kanehisa2002] M. Kanehisa, S. Goto, S. Kawashima and A. Nakaya: «*The KEGG databases at GenomeNet*» (2002) **Nucleic Acids Research**, 30, Seiten 2–46

[Khan1998] J. Khan, R. Simon, M. Bittner, Y. Chen, S. B. Leighton, T. Pohida, P. D. Smith, Y. Jiang, G. C. Gooden, J. M. Trent, P. S. Meltzer: «*Gene expression profiling of alveolar rhabdomyosarcoma with cDNA microarrays*» (1998) **Cancer Research**, 58, Seiten 5009–5013

[Kim2003] J.D. Kim, T. Ohta, Y. Tateisi, J. Tsujii: «*GENIA corpus-a semantically annotated corpus for bio-textmining*» (2003 Jul) **Bioinformatics**, Band 19, Suppl. 1, Seiten 180-182

[Knudsen2003] S. Knudsen, C. Workman, T. Sicheritz-Ponten and C. Friis: «*GenePublisher automated analysis of DNA microarray data*» (2003) **Nucleic Acids Research**, Band 31, 13, Seiten 3471–3476

[Kohlmann2003] A. Kohlmann, C. Schoch, S. Schnittger, M. Dugas, W. Hiddemann, W. Kern, T. Haferlach: «*Molecular characterization of acute leukemias by use of microarray technology*» (2003 Aug) **Genes Chromosomes Cancer**, Band 37, 4, Seiten 396-405

[Krone1999] A.Krone: «*Datenbasiertefdf Generierung von relevanten Fuzzy-Regeln zur Modellierung von Prozesszusammenhangen und Bedienstrategien*» (1999) **Fortschritt-Berichte VDI**, Reihe 10, Nr. 615, Vdi Verlag, Düsseldorf.

[Krull2003] M.Krull, N.Voss, S.Choi, S. Pistor, A.Potapov and E.Wingender: «*TRANSPATH: an integrated database on signal transduction and a tool for array analysis*» (2003) **Nucleic Acids Research**, 31, Seiten 97–100

[Lane1999] D. Lane: HyperStat Online Textbook. Chapter 16 Chi Square (1999)

[Langley1992] P.Langley,W. Iba and K. Thompson: «*An analysis of Bayesian classifier*» (1992), Proceedings of the Tenth National Confer-ence on Artificial Intelligence. AAAI Press, New York, Seiten 223-228

[Lawrence1999] H.J. Lawrence, S. Rozenfeld, C. Cruz, K. Matsukuma, A. Kwong, L. Komuves, A.M. Buchberg, C. Largman: «*Frequent co-expression of the HOXA9 and MEIS1 homeobox genes in human myeloid leukemias*» (1999 Dec) **Leukemia**, Band 13, 12, Seiten 1993-1999

[Lee2002] Y. Lee and C. Lee: «*Classification of multiple cancer types by multicategory support vector machines using gene expression data*» (2002) Technical report 1051., Madison Wl University of Wisconsin Department of Statistics

[Li2002] Y. Li, C. Campbell, M. Tipping: «*Bayesian automatic relevance determination algorithms for classifying gene expression data*» (2002 Oct) **Bioinformatics**, Band 18, 10, Seiten 1332-1339

[Li2002] J. Li, L. Wong: «*Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns*» (2002 May) **Bioinformatics**, Band 18, 5, Seiten 725-734

[Liu2003] G.Liu, A.E.Loraine, R.Shigeta, M.Cline,J. Cheng, V.Valmeekam,S.Sun, D.Kulp and M.A. Siani-Rose: «*NetAffx: Affymetrix probesets and annotations*» (2003) **Nucleic Acids Research**, 31, Seiten 82–86

[Lockhart1996] D. J. Lockhart, H. L. Dong, M. C. Byrne, M. T. Follettie, M. V. Gallo, M. S. Chee, M. Mittmann, C. Wang, M. Kobayashi, and H. Horton: «*Expression monitoring by hybridization to high-density oligonucleotide arrays*» (1996) **Nature Biotechnology**, 14, Seiten 675-1680

[Loh1997] W.-Y. Loh and Y.-S, Shih: „*Split selection methods for classification trees*“. (1997), **Statistica Sinica**, Band 7, Seiten 815-840.

[Lucenko1999] E.V.Lutsenko, V.S.Simankov : «*Adaptive control of complex systems on the basis of the pattern recognition theory*» (1999), Krasnodar, Technical specifications KUBGU

[Mack2002] R. Mack, M. Hehenberger: «*Text-based knowledge discovery search and mining of life-sciences documents*» (2002 Jun 1) **Drug Discov Today**, Band 7, Suppl.11, Seiten 89-98

[Mavilio1986] F. Mavilio, N.M. Sposi, M. Petrini, L. Bottero, M. Marinucci, G. De Rossi, S. Amadori, F. Mandelli, C. Peschle: «*Expression of cellular oncogenes in primary cells from human acute leukemias*» (1986 Jun) **Proc Natl Acad Sci U S A**, Band 83, 12, Seiten 4394-4398

[McLachlan1992] G. J. McLachlan: «*Discriminant analysis and statistical pattern recognition*» (1992), Wiley, New York

[Mitchell1997] T. Mitchell: «*Machine Learning*» (1997), The McGraw-Hill Companies Inc.

[Nguyen2002a] D.V. Nguyen, D.M. Rocke: «*Multi-class cancer classification via partial least squares with gene expression profiles*» (2002 Sep) **Bioinformatics**, Band 18, 9, Seiten 1216-1226

[Nguyen2002b] D.V. Nguyen, D.M. Rocke: «*Tumor classification by partial least squares using microarray gene expression data*» (2002 Jan) **Bioinformatics**, Band 18, 1, Seiten 39-50

[Ochoa] J. A. C. Ochoa, J. Ruiz-Shulcloper, Lucia de la Vega Doria: «*Fuzzy KORA-Q algorithm*» (1998), Proceedings of the Sixth European Congress on Intelligent Techniques and Soft Computing (EUFIT'98) Aachen Germany, Seiten 1190-1194

[Ohrn1998] A. Ohrn, L. Ohno-Machado, T. Rowland: «*Building manageable rough set classifiers*» (1998) Proc AMIA Symp, Seiten 543-547

[Okita2000] H. Okita, A. Umezawa, M. Fukuma, T. Ando, F. Urano, M. Sano, Y. Nakata, T. Mori, J. Hata: «*Acute myeloid leukemia possessing jumping translocation is related to highly elevated levels of EAT/mcl-1, a Bcl-2 related gene with anti-apoptotic functions*» (2000 Jan) **Leuk Res**, Band 24, 1, Seiten 73-77

[Oyama2002] T. Oyama, K. Kitano, K. Satou, T. Ito: «*Extraction of knowledge on protein-protein interaction by association rule discovery*» (2002) **Bioinformatics**, Band 18, 5, Seiten 705-714

[Page2000] W. Page: «*Using ORACLE8/8i*» (2000), Special Edition, **QUE Corporation.**

[Park2001] P Park, M Pagano, M Bonetti : «*A nonparametric scoring algo-rithm for identifying informative genes from microarray data*» (2001) **Pac Symp Biocomput**, 6

[Perou1999] C.M.Perou, S.S.Jeffrey, M.van de Rijn, C.A.Rees, M.B.Eisen, D.T.Ross, A. Pergamenschikov, C.F.Williams, S.X.Zhu, J.C.Lee F, D.Lashkari, D.Shalon, P.O.Brown and D.Botstein : «*Distinctive gene expression patterns in human mam-mary epithelial cells and breast cancers*» (1999) **Proc Natl Acad Sci U S A**, 96, Seiten 9212-9217

[Poljakova1976] M.P.Poljakova, M.N.Vajntsvajg: «*About a usage of voting method in recognition algorithms(in russ.)*» (1976), in "Applied problems of learning", Moscow, Seiten 25-28

[Pomeroy2002] S. Pomeroy, P. Tamayo, M.Gaasenbeek, L. Sturla, M. Angelo, M. McLaughlin, J. Kim, L. Goumnerova, P. Black, C. Lau et al.: «*Prediction of central nervous system embryonal tumor outcome based on gene expression*» (2002) **Nature**, 415, Seiten 36-442

[Pitts2000] N. Pitts: «*XML in Record Time*» (2000) SYBEX Inc.

[Pustejovsky2001] J. Pustejovsky, J. Castano, B. Cochran, M. Kotecki, M. Morrell: «*Automatic extraction of acronym-meaning pairs from MEDLINE databases*» (2001) **Medinfo**, Band 10, Pt 1, Seiten 371-375

[Quinlan1986] R. Quinlan: «*Induction of decision trees*» (1986) **Machine Learning**, 1, Seiten 101–106

[Quinlan1993] R. Quinlan: «*C4.5: Programs for Machine Learning*» (1993), Morgan Kaufmann, San Mateo, CA

[Reuther2002] G.W. Reuther, Q.T. Lambert, J.F. Rebhun, M.A. Caligiuri, L.A. Quilliam, C.J. Der: «*RasGRP4 is a novel Ras activator isolated from acute myeloid leukemia*» (2002 Aug 23) **J Biol Chem**, Band 277, 34, Seiten 30508-30514

[Ripley1996] D. Ripley: «*Pattern recognition and neural networks*» (1996), Cambridge University Press, Cambridge, New York

[Robson2003] B. Robson: «*Clinical and pharmacogenomic data mining 1. Generalized theory of expected information and application to the development of tools*» (2003 May-Jun) **J Proteome Res**, Band 2, 3, Seiten 283-302

[Ross2000] T. Ross, U. Scherf, M. B. Eisen, C. M. Perou, P. Spellman, V. Iyer, S. S. Jeffrey, M. Van de Rijn, M. Waltham, A. Pergamenschikov, J. C. F. Lee, D. Lashkari, D. Shalon, T. G. Myers, J. N.

Weinstein, D. Botstein, and P. O. Brown: «*Systematic variation in gene expression patterns in human cancer cell lines*» (2000) **Nature Genetics**, 24, Seiten 227-234

[Ruiz-Shulcloper] J. Ruiz-Shulcloper: «*Logical Combinatorial Pattern Recognition*» (2000 September), Foro Iberoamericano de Reconocimiento de Patrones, Barcelona Spain, Seiten 128-138

[Sarkar2000] M. Sarkar, TY Leong: «*Application of K-nearest neighbors algorithm on breast cancer diagnosis problem*» (2000) Proc AMIA Symp, Seiten 759-763

[Schena1995] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown.: «*Quantitative monitoring of gene expression patterns with a complementary dna microarray*» (1995) **Science**, 270, Seiten 467-470

[Schena1999] M. Schena: «*DNA Microarrays*» (1999), A Practical Approach. Oxford University Press

[Schuchhardt2000] J. Schuchhardt, D. Beule, A. Malik, E. Wolski, H. Eickhoff, H. Lehrach, H. Herzel: «*Normalization strategies for cDNA microarrays*» (2000) **Nucleic Acids Research**, 28 (10),

[Schuster2003] S. Schuster: «*Metabolic pathway analysis in biotechnology*» (2003), In Metabolic Engineering in the Post Genomic Era, Horizon Scientific, Wymondham, Seiten 181-208

[Schuurmans1999] D. Schuurmans: «*Machine Learning course notes*» (1999), University of Waterloo

[Shiffman1994] R.N. Shiffman, R.A. Greenes: «*Improving clinical guidelines with logic and decision-Tabelle techniques application to hepatitis immunization recommendations*» (1994 Jul-Sep) **Med Decis Making**, Vol14, 3, Seiten 245-254

[Singh2003] S.B. Singh, R.D. Hull, E.M. Fluder: «*Text Influenced Molecular Indexing (TIMI) a literature database mining approach that handles text and chemistry*» (2003 May-Jun) **J Chem Inf Comput Sci**, Band 43, 3, Seiten 743-752

[StatSoft1998] StatSoft (1998), Electonic Textbook StatSoft Inc.

[Stephenson1995] J. Stephenson, H. Lizhen, G.J. Mufti: «*Possible co-existence of RAS activation and monosomy 7 in the leukämic transformation of myelodysplastic syndromes*» (1995 Oct) **Leuk Res**, Band 19, 10, Seiten 741-748

[Sun1991] G. Sun, S. Wormsley, R.S. Sparkes, F. Näim, R.P. Gale: «*Hybrid leukemia and the 5q-abnormality*» (1991) **Leuk Res**, Band 15, 5, Seiten 351-356

[Szabo2002] A. Szabo, K. Boucher, W.L. Carroll, L.B. Klebanov, A.D. Tsodikov, A.Y. Yakovlev: «*Variable selection and pattern recognition with gene expression data generated by the microarray technology*» (2002 Mar) **Math Biosci**, Band 176, 1, Seiten 71-98

[Thomas2001] J.G. Thomas, J.M. Olson, S.J. Tapscott and L.P. Zhao: «*An efficient and robust statistical modeling approach to discover differentially expressed genes using genomic expression profiles*» (2001 Jul) **Genome Res**, Band 11, 7, Seiten 1227-1236

[Tibshirani1999] R. Tibshirani, T. Hastie, M. Eisen, D. Ross, D. Botstein and P. Brown: «*Clustering methods for the analysis of dna microarray data*» (1999) Technical report, Department of Health Research and Policy Stanford University

[Troussard1995] X. Troussard, R. Rimokh, F. Valensi, D. Leböuf, O. Fenneteau, A.M. Guitard, A.M. Manel, F. Schillinger, C. Leglise and A. Brizard: «*Heterogeneity of t(1, 19)(q23, p13) acute leukämias. French Hämatological Cytology Group*» (1995 Mar) **Br J Hämatol**, Band 89, 3, Seiten 516-526

[Vapnik2000] V. N. Vapnik: «*The nature of statistical learning theory*» (2000), 2nd edition, Statistics for engineering and information science, Springer, New York

[Velde1989] V. de Velde: «IDL: Taming the Multiplexer» (1989)

[Viator2001] J.A. Viator, F.M. Pestorius: «*Investigating trends in acoustics research from 1970-1999*» (2001 May) J Acoust Soc Am, Band 109, 5 Pt 1, Seiten 1779-1783

[Weber2000] J. Weber: «*Using Java*» (2000) Special Edition, **QUE Corporation.**

[Winston1992] P.H Winston: «*Artificial Intelligence*» (1992), Third Edition, Addison-Wesley

[Yeung2001] K.Y.Yeung, D.R.Haynor and W.L.Ruzzo: «*Validating clustering for gene expression data*» (2001) **Bioinformatics**, 17, Seiten 309–318

[Zeng1997] C. Zeng, A.J. van Wijnen, J.L. Stein, S. Meyers, W. Sun, L. Shopland, J.B. Lawrence, S. Penman, J.B. Lian, G.S. Stein and S.W. Hiebert: «*Identification of a nuclear matrix targeting signal in the leukemia and bone-related AML/CBF-alpha transcription factors*» (1997 Jun 24) **Proc Natl Acad Sci U S A**, Band 94, 13, Seiten 6746-6751

[Zhang1997] L. Zhang, W. Zhou, V. E. Velculescu, S. E. Kern, R. H. Hruban, S. R. Hamilton, B. Vogelstein, K. W. Kinzler: «*Gene expression profiles in normal and cancer cells*» (1997) **Science**, 276 (5316), Seiten 1268–1272

[Zien2001] A. Zien, T. Aigner, R. Zimmer and T. Lengauer: «*Centralization: a new method for the normalization of gene expression data*» (2001) **Bioinformatics**, Band 17, Suppl. 1

ANHANG A

8. Ein Beispiel der Regelanalysis

Der Datensatz besteht aus 16 Männergesichtern (Abb. 8.1), die zu 2 Klassen gehören. Die Gesichter mit der Nummer 1 bis 8 in Abbildung 8.1 oben gehören zur Klasse 1 und die unteren acht Gesichter zur zweiten Klasse. Auf der Basis der existierenden Objekteigenschaften der Gesichter werden logische Regeln erzeugt, die eine Klassifikation der Männergesichter gestattet.

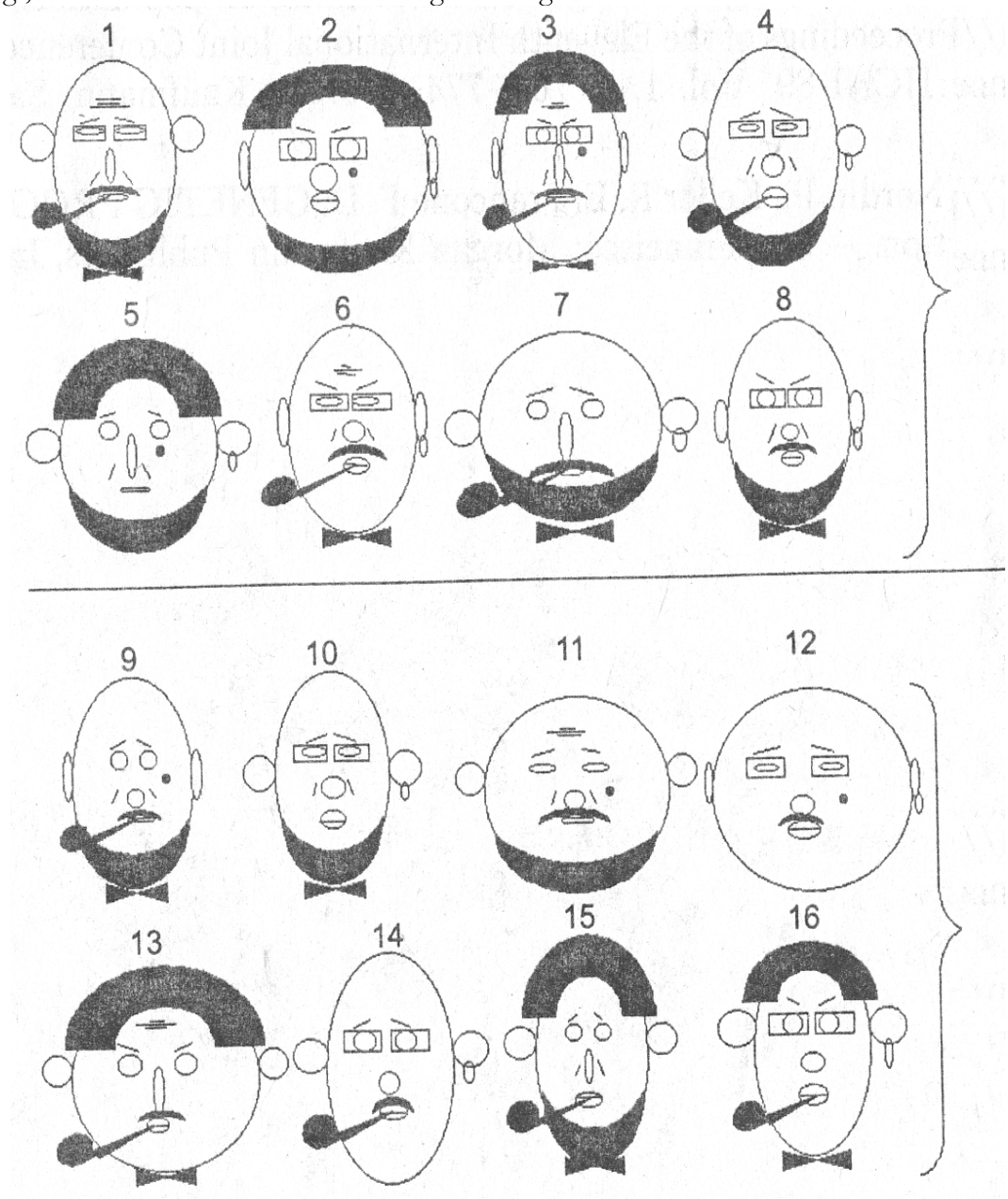

Abbildung 8.1. Datensatz „Gesichter“. Oben ist die erste Klasse und unten ist die zweite Klasse

Die Erzeugung der logischen Regeln erfolgt in verschiedenen Etappen. Im ersten Schritt werden die 16 existierenden Objekteigenschaften binär verschlüsselt (Tab. 8.1). Eine Eins bedeutet, dass die entsprechende Objekteigenschaft vorhanden ist und eine Null, dass die Eigenschaft für das entsprechende Objekt nicht zutrifft.

	Beschreibung	Wert	Gesichter der 1.Klasse								Gesichter der 2.Klasse							
			#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16
X_1	Der runde Kopf	Ja =1;Nein =0	0	1	0	0	1	0	1	0	0	0	1	1	1	0	0	0
X_2	Die angedrückte Ohren	Ja =1;Nein =0	1	0	0	1	1	0	1	0	0	1	1	0	1	1	1	1
X_3	Die runde Nase	Ja =1;Nein =0	0	1	0	1	0	1	0	1	1	1	1	1	0	1	0	1
X_4	Die runden Augen	Ja =1;Nein =0	0	1	1	0	1	0	1	1	1	0	0	0	1	1	1	1
X_5	Die runzelige Stirn	Ja =1;Nein =0	1	0	1	0	0	1	0	0	0	0	1	1	1	0	0	0
X_6	Naselippen Falte	Ja =1;Nein =0	1	0	1	1	1	1	0	1	1	1	1	0	0	0	1	0
X_7	Die dicken Lippen	Ja =1;Nein =0	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1
X_8	Die Haare	Ja =1;Nein =0	0	1	1	0	1	0	0	0	0	0	0	0	1	0	1	1
X_9	Die Schnurrbärte	Ja =1;Nein =0	1	0	1	0	0	1	1	1	1	0	1	1	1	1	0	0
X_{10}	Der Bart	Ja =1;Nein =0	1	1	0	1	1	0	1	1	1	1	1	0	0	0	1	0
X_{11}	Die Brillen	Ja =1;Nein =0	1	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1
X_{12}	Der Naevus	Ja =1;Nein =0	0	1	1	0	1	0	0	0	1	0	1	1	0	0	0	0
X_{13}	Der Schmetterling	Ja =1;Nein =0	1	0	1	0	0	1	1	1	1	1	0	0	1	0	1	1
X_{14}	Die Augenbrauen nach oben	Ja =1;Nein =0	1	0	0	1	1	0	1	0	1	1	1	1	0	1	1	0
X_{15}	Der Ohrring	Ja =1;Nein =0	0	1	0	1	1	1	1	1	0	1	0	1	0	1	0	1
X_{16}	Die Pfeife	Ja =1;Nein =0	1	0	1	1	0	1	1	0	1	0	0	0	1	1	1	1

Tabelle 8.1. „Gesichter“ nach der Kodierung.

Im nachfolgenden Text wird die Erzeugung von logischen Regeln für die Objekte der Klasse 1 ausführlich erläutert. Der Algorithmus wird mit folgenden Konfigurationswerten parametrisiert (Tab. 8.2).

Parameter		Wert
Min. Richtig		4
Max.Fehler		0
Einschätzung der Qualität der neuen logischen Regel	Min.Schritt	1
	Konjunktionsrange	unbegrenzt
Merkmalssortierung		Sortierung der falschen Erkennungen

Tabelle 8.2. Konfigurationswerten des Algorithmus.

In der Tabelle 8.3 ist die Anzahl der Objekte entsprechend der Objekteigenschaft und der Klasse aufgeführt. Weil jeden der Merkmalenwerte binär („ja“ oder „nein“) ist, haben wir 32 Merkmale. Da jede Objekteigenschaft binär codiert ist, ergeben sich insgesamt 32 Möglichkeiten (Merkmale) der Klassifikation. Keine der Merkmale kann die beiden Klassen allein teilen, deshalb muss nach Kombinationen von Merkmalen gesucht werden.

Wert	Klasse	Variablen															
		X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}
1	1	3	4	4	5	3	6	4	3	5	6	6	3	5	4	6	5
	2	3	6	6	5	3	4	6	3	5	4	4	3	5	6	4	5
0	1	5	4	4	3	5	2	4	5	3	2	2	5	3	4	2	3
	2	5	2	2	3	5	4	2	5	3	4	4	5	3	2	4	3

Tabelle 8.3. Die Anzahl der Objekte entsprechend Objekteigenschaft und der Klasse.

Im nächsten Schritt werden anhand des Parameters „Min. Richtig“ die Merkmale ausgewählt, die keine Klassifikation der Objekte gestatten. Diese Merkmale werden in den nachfolgenden Schritten der Regelgenerierung nicht mehr berücksichtigt. Für die Objekte der Klasse 1 sind es die Merkmale X_1 =1, X_4=0, X_5=1, X_6=0, X_8=1, X_9=0, X_{10}=0, X_{11}=0, X_{12}=1, X_{13}=0, X_{15}=0 und X_{16}=0.

Für die Sortierung der Merkmale existieren verschiedene Varianten (siehe Beschreibung des Verfahrens). Für die Bearbeitung des vorliegenden Fallbeispiels wird die Sortierung „der falschen Erkennung“ (die Menge der Objekte, die zu abtrennen sind) verwendet.

Bei gleicher Fehlerzahl werden die Merkmale nach der Anzahl der richtigen Erkennungen in der ersten Klasse sortiert. Die erhaltenen Ergebnisse sind in der Tabelle 8.4 dargestellt. Für die nachfolgenden Analysenschritte werden die Merkmale entsprechend ihrer Sortierreihenfolge (erste Spalten zuerst!) verwendet.

	Variablen																			
Merkmal	X_2	X_3	X_7	X_{14}	X_6	X_{10}	X_{11}	X_{15}	X_1	X_4	X_5	X_8	X_9	X_{12}	X_{13}	X_{16}	X_2	X_3	X_7	X_{14}
Wert	0	0	0	0	1	1	1	1	0	1	0	0	1	0	1	1	1	1	1	1
In der Klasse 1	4	4	4	4	6	6	6	6	5	5	5	5	5	5	5	5	4	4	4	4
In der Klasse 2	2	2	2	2	4	4	4	4	5	5	5	5	5	5	5	5	6	6	6	6

Tabelle 8.4. Die reduzierte und sortierte Objektanzahl, entsprechend der Objekteigenschaft und der Klasse

Am Anfang der Liste befindet sich das Merkmal X_2=0. Für dieses Merkmal werden alle gültigen Kombinationen gesucht. Das erhaltene Ergebnis ist in Tabelle 8.5 dargestellt. Nur die Merkmale X_{14} = 0 und X_{11} = 1 befriedigen die erste Bedingung und können für die weitere Arbeit verwendet werden.

Das Merkmal X_{14} = 0 ermöglicht die Klasse 1 von der Klasse 2 zu unterscheiden. Die Konjunktion

„X_2=0 und X_{14} = 0“ (1)

wird protokolliert. Das Merkmal X_{11} = 1 ermöglicht keine weitere Differenzierung zwischen den Klassen. Nach dem Abschluss der Analyse für das Merkmal X_2=0 wird es im weiteren Verfahren nicht mehr berücksichtigt. Danach wird die Analyse mit dem nächsten Merkmal X_3 =0 fortgesetzt. Nur das Merkmal X_7 = 0ermöglicht eine Unterscheidung zwischen den Klassen. Die Konjunktion

„X_3=0 und X_7 = 0“ (2)

wird protokolliert und die Analyse wird mit dem folgenden Merkmal X_7 = 0 (siehe Tabelle 8.4) fortgesetzt. Für dieses Merkmal ist es nicht möglich eine gültige Konjunktion zu finden, die eine Klassifikation ermöglicht. Die gleiche Aussage gilt auch für darauf folgende Merkmal X_{14} = 0 (siehe Tabelle 8.4).

	Objekte					
#	2	3	6	8	9	12
Klasse	1	1	1	1	2	2
X_1		0	0	0	0	
X_3	1	0	1	1	1	1
X_4	1	1		1	1	
X_5	0			0	0	
X_6		1	1	1	1	
X_7	1	0	1	1	0	1
X_8			0	0	0	0
X_9		1	1	1	1	1
X_{10}	1			1	1	
X_{11}	**1**	**1**	**1**	**1**		**1**
X_{12}			0	0		
X_{13}		1	1	1	1	
X_{14}	**0**	**0**	**0**	**0**	**1**	**1**
X_{15}	1		1	1		1
X_{16}		1	1		1	

Tabelle 8.5. Kombinationenanalyse der Konjunktion X_2=0. In der Tabelle sind die Bedeutengen (0/1) der Variablen X_1 - X_{16} von Objekten (#) beider Klassen (1/2), die durch diese Konjunktion (X_2=0) erkannt sind, dargestellt. Die Variablen, die für die weitere Bearbeitung interessant sind, sind graufarbig markiert.

	Objekte					
#	1	3	5	7	13	15
Klasse	1	1	1	1	2	2
X_1	0	0				0
X_2	1		1	1	1	1
X_4		1	1	1	1	1
X_5			0	0		0
X_6	1	1	1			1
X_7	**0**	**0**	**0**	**0**	**1**	**1**
X_8	0			0		
X_9	1	1		1	1	
X_{10}	1		1	1		1
X_{11}	1	1				
X_{12}	0			0	0	0
X_{13}	1	1		1	1	1
X_{14}	1	0	1	1	0	1
X_{15}			1	1		
X_{16}	1	1		1	1	1

Tabelle 8.6. Kombinationenanalyse der Konjunktion X_3=0. (s. Tab.8.5).

	Objekte					
#	1	3	5	7	9	11
Klasse	1	1	1	1	2	2
X_1	0	0			0	
X_2	1		1	1		1
X_3					1	1
X_4		1	1	1	1	
X_5			0	0	0	
X_6	1	1	1		1	1
X_8	0			0	0	0
X_9	1	1		1	1	1
X_{10}	1		1	1	1	1
X_{11}	1	1				
X_{12}	0			0		
X_{13}	1	1		1	1	
X_{14}	1	0	1	1	1	1
X_{15}			1	1		
X_{16}	1	1		1	1	

Tabelle 8.7. Kombinationenanalyse der Konjunktion X_7 = 0. (s. Tab.8.5).

	Objekte					
#	2	3	6	8	13	16
Klasse	1	1	1	1	2	2
X_1		0	0	0		0
X_2					1	1
X_3	1		1	1		1
X_4	1	1		1	1	1
X_5	0			0		0
X_6		1	1	1		
X_7	1		1	1	1	1
X_8			0	0		
X_9		1	1	1	1	
X_{10}	1			1		
X_{11}	**1**	**1**	**1**	**1**		**1**
X_{12}			0	0	0	0
X_{13}		1	1	1	1	1
X_{15}	1		1	1		1
X_{16}		1	1		1	1

Tabelle 8.8. Kombinationenanalyse der Konjunktion X_{14} = 0. (s. Tab.8.5).

Das nächste zu analysierende Merkmal ist $X_6 = 1$. Die erzielten Ergebnisse für das Merkmal $X_6 = 1$ sind in den Tabellen 8.9 und 8.10 dargestellt.

	Objekte									
#	1	3	4	5	6	8	9	10	11	15
Klasse	1	1	1	1	1	1	2	2	2	2
X_1	**0**	**0**	**0**		**0**	**0**	**0**	**0**		**0**
X_2	1		1	1				1	1	1
X_3			1		1	1	1	1	1	
X_4		1		1		1	1			1
X_5			0	0		0	0	0		0
X_7			1		1	1		1		1
X_8	**0**		**0**		**0**	**0**	**0**	**0**	**0**	
X_9	**1**	**1**			**1**	**1**	**1**		**1**	
X_{10}	1		1	1		1	1	1	1	1
X_{11}	**1**	**1**	**1**		**1**	**1**		**1**		
X_{12}	**0**		**0**		**0**	**0**		**0**		**0**
X_{13}	**1**	**1**			**1**	**1**	**1**	**1**		**1**
X_{14}	1		1	1			1	1	1	1
X_{15}			**1**	**1**	**1**	**1**		**1**		
X_{16}	**1**	**1**	**1**		**1**		**1**			**1**

Tabelle 8.9. Kombinationenanalyse der Konjunktion $X_6 = 1$. (s. Tab.8.5).

Merkmal	X_{11}	X_{15}	X_9	X_{12}	X_{16}	X_1	X_8	X_{13}
Wert	1	1	1	0	1	0	0	1
In der Klasse 1	5	4	4	4	4	5	4	4
In der Klasse 2	1	1	2	2	2	3	3	3

Tabelle 8.10. Die Anzahl der Objekte (Gesichter) entsprechend der Objekteigenschaft und der Klasse für $X_6 = 1$

Für dieses Merkmal existiert keine einfache Konjunktion, die eine Unterscheidung zwischen den beiden Klassen ermöglicht. Die Merkmale $X_2 – X_5$, X_7, X_{14} werden bei der weiteren Analyse nicht berücksichtigt, weil sie die erste Bedingung nicht erfüllen, und das Merkmal X_{10} befriedigt nicht die dritte Bedingung. Die übrigen Merkmale werden sortiert.

Für «$X_6 = 1$ und $X_{11}= 1$» können $X_9 =1$ und $X_{16}= 1$ die Klassen aufteilen. Die Konjunktionen

- «$X_6 = 1$ und $X_{11}= 1$ und $X_9= 1$» (3)
- «$X_6 = 1$ und $X_{11}= 1$ und $X_{16}= 1$» (4)

werden protokolliert. Für die übrigen Merkmale ergeben sich keine Regeln.

	Objekte					
#	1	3	4	6	8	10
Klasse	1	1	1	1	1	2
X_1	0	0	0	0	0	0
X_8	0		0	0	0	0
X_9	**1**	**1**		**1**	**1**	
X_{12}	0		0	0	0	0
X_{13}	1	1		1	1	1
X_{15}			1	1	1	1
X_{16}	**1**	**1**	**1**	**1**		

Tabelle 8.11. Kombinationenanalyse der Konjunktion $X_6 = 1$ und $X_{11}= 1$. (s. Tab.8.5).

	Objekte				
#	4	5	6	8	10
Klasse	1	1	1	1	2
X_1	0		0	0	0
X_8	0		0	0	0
X_9			1	1	
X_{12}	0		0	0	0
X_{13}			1	1	1
X_{16}	1		1		

Tabelle 8.12. Kombinationenanalyse der Konjunktion $X_6 = 1$ und $X_{15}= 1$. (s. Tab.8.5).

	Objekte					
#	1	3	6	8	9	11
Klasse	1	1	1	1	2	2
X_1	0	0	0	0	0	
X_8	0		0	0	0	0
X_{12}	0		0	0		
X_{13}	1	1	1	1	1	
X_{16}	1	1	1		1	

Tabelle 8.13. Kombinationenanalyse der Konjunktion $X_6 = 1$ und $X_9= 1$. (s. Tab.8.5).

	Objekte					
#	1	4	6	8	10	15
Klasse	1	1	1	1	2	2
X_1	0	0	0	0	0	0
X_8	0	0	0	0	0	
X_{13}	1		1	1	1	1
X_{16}	1	1	1			1

Tabelle 8.14. Kombinationenanalyse der Konjunktion $X_6 = 1$ und $X_{12}= 0$. (s. Tab.8.5).

	Objekte					
#	1	3	4	6	9	15
Klasse	1	1	1	1	2	2
X_1	0	0	0	0	0	0
X_8	0		0	0	0	
X_{13}	1	1		1	1	1

Tabelle 8.15. Kombinationenanalyse der Konjunktion $X_6 = 1$ und $X_{16}= 1$. (s. Tab..8.5).

	Objekte							
#	1	3	4	6	8	9	10	15
Klasse	1	1	1	1	1	2	2	2
X_8	0		0	0	0	0	0	
X_{13}	1	1		1	1	1	1	1

Tabelle 8.16. Kombinationenanalyse der Konjunktion $X_6 = 1$ und $X_1= 0$. (s. Tab.8.5).

	Objekte							
#	1	4	6	8	9	10	11	
Klasse	1	1	1	1	2	2	2	
X_{13}	1		1	1	1	1		

Tabelle 8.17. Kombinationenanalyse der Konjunktion $X_6 = 1$ und $X_8= 0$. (s. Tab.8.5).

Für das folgende Merkmal $X_{10} =1$ existiert keine einfache Konjunktion Die Merkmale X_1, X_3, X_7, X_9, X_{13}, X_{16} können nicht berücksichtigt werden, weil sie die erste Bedingung nicht erfüllen, und X_{14} befriedigt nicht die dritte Bedingung. Die übrigen Merkmale (die Tabellen) werden sortiert. Die Ergebnisse für die Merkmale sind in den Tabellen 8.18 und 8.19 dargestellt.

	Objekte									
#	1	2	4	5	7	8	9	10	11	15
Klasse	1	1	1	1	1	1	2	2	2	2
X_1	0		0			0	0	0		0
X_2	**1**		**1**	**1**	**1**			**1**	**1**	**1**
X_3		1	1			1	1	1	1	
X_4		**1**		**1**	**1**	**1**	**1**			**1**
X_5		**0**	**0**	**0**	**0**	**0**	**0**	**0**		**0**
X_7		1	1			1		1		1
X_8	**0**		**0**		**0**	**0**	**0**	**0**	**0**	
X_9	1				1	1	1		1	
X_{11}	**1**	**1**	**1**			**1**		**1**		
X_{12}	**0**		**0**		**0**	**0**		**0**		**0**
X_{13}	1				1	1	1	1		1
X_{14}	1		1	1	1		1	1	1	1
X_{15}		**1**	**1**	**1**	**1**	**1**		**1**		
X_{16}	1		1		1		1			1

Tabelle 8.18. Kombinationenanalyse der Konjunktion $X_{10} = 1$. (s. Tab.8.5).

Merkmal	X_{15}	X_{11}	X_4	X_{12}	X_5	X_8	X_2
Wert	1	1	1	0	0	0	1
In der Klasse 1	5	4	4	4	5	4	4
In der Klasse 2	1	1	2	2	3	3	3

Tabelle 8.19. Die Anzahl der Objekte (Gesichter) entsprechend der Objekteigenschaft und der Klasse für $X_{10} = 1$

Für «$X_{10} = 1$ und $X_{15}= 1$» kann $X_4 =1$ die Klassen aufteilen. Die Konjunktion

«$X_{10} = 1$ und $X_{15}= 1$ und $X_4 =1$» (5)

wird protokolliert. Für die übrigen Merkmale gibt es keine Lösung.

	Objekte					
#	2	4	5	7	8	10
Klasse	1	1	1	1	1	2
X_2		1	1	1		1
X_4	**1**		**1**	**1**	**1**	
X_5	0	0	0	0	0	0
X_8		0		0	0	0
X_{11}	1	1			1	1
X_{12}		0		0	0	0

Tabelle 8.20. Kombinationenanalyse der Konjunktion $X_{10} = 1$ und $X_{15} = 1$. (s. Tab.8.5).

	Objekte				
#	1	2	4	8	10
Klasse	1	1	1	1	2
X_2	1		1		1
X_4		1		1	
X_5		0	0	0	0
X_8	0		0	0	0
X_{12}	0		0	0	0

Tabelle 8.21. Kombinationenanalyse der Konjunktion $X_{10} = 1$ und $X_{11} = 1$. (s. Tab.8.5).

	Objekte					
#	2	5	7	8	9	15
Klasse	1	1	1	1	2	2
X_2		1	1			1
X_5	0	0	0	0	0	0
X_8			0	0	0	
X_{12}			0	0		0

Tabelle 8.22. Kombinationenanalyse der Konjunktion $X_{10} = 1$ und $X_4 = 1$. (s. Tab.8.5).

	Objekte					
#	1	4	7	8	10	15
Klasse	1	1	1	1	2	2
X_2	1	1	1		1	1
X_5		0	0	0	0	0
X_8	0	0	0	0	0	

Tabelle 8.23. Kombinationenanalyse der Konjunktion $X_{10} = 1$ und $X_{12} = 0$. (s. Tab.8.5).

	Objekte							
#	2	4	5	7	8	9	10	15
Klasse	1	1	1	1	1	2	2	2
X_2		1	1	1			1	1
X_8		0		0	0	0	0	

Tabelle 8.24. Kombinationenanalyse der Konjunktion $X_{10} = 1$ und $X_5 = 0$. (s. Tab.8.5).

	Objekte						
#	1	4	7	8	9	10	11
Klasse	1	1	1	1	2	2	2
X_2	1	1	1			1	1

Tabelle 8.25. Kombinationenanalyse der Konjunktion $X_{10} = 1$ and $X_8 = 0$. (s. Tab.8.5).

Das nächste zu analysierende Merkmal ist $X_{11} =1$. Für dieses Merkmal existiert keine einfache Konjunktion, die eine Aufteilung der Klassen ermöglicht. Die Analyse wird mit der Suche nach gültigen Konjunktionen fortgesetzt. Die Merkmale X_2, X_4, X_5, X_{14} können nicht berücksichtigt, weil sie die erste Bedingung nicht befriedigen, und die Merkmale X_3, X_7, X_{15} erfüllen nicht die zweite Bedingung. Die übriggebliebenen Merkmale (siehe Tabelle) werden sortiert. Die Ergebnisse für das Merkmal $X_{11} =1$sind in den Tabellen 8.26 und 8.27 dargestellt.

Für «$X_{11} = 1$ und $X_9= 1$» kann $X_{13} =1$ die Klassen aufteilen. Die Konjunktion

«$X_{11} = 1$ und $X_9= 1$ und $X_{13} =1$» (6)

wird protokolliert. Für die übrigen Merkmale gibt es keine Lösung.

	Objekte									
#	1	2	3	4	6	8	10	12	14	16
Klasse	1	1	1	1	1	1	2	2	2	2
X_1	0		0	0	0	0	0		0	0
X_2	1			1			1		1	1
X_3		1		1	1	1	1	1	1	1
X_4		1	1			1			1	1
X_5		0		0		0	0		0	0
X_7		1		1	1	1	1	1	1	1
X_8	0			0	0	0	0	0	0	
X_9	1		1		1	1		1	1	
X_{12}	0			0	0	0	0		0	0
X_{13}	1		1		1	1	1			1
X_{14}	1			1			1	1	1	
X_{15}		1		1	1	1	1	1	1	1
X_{16}	1		1	1	1				1	1

Tabelle 8.26. Kombinationenanalyse der Konjunktion $X_{11} = 1$. (s. Tab.8.5).

Merkmal	X_9	X_{13}	X_{16}	X_1	X_8	X_{12}
Wert	1	1	1	0	0	0
In der Klasse1	4	4	4	5	4	4
In der Klasse 2	2	2	2	3	3	3

Tabelle 8.27. Die Anzahl der Objekte (Gesichter) entsprechend der Objekteigenschaft und der Klasse für $X_{11} = 1$

	Objekte					
#	1	3	6	8	12	14
Klasse	1	1	1	1	2	2
X_1	0	0	0	0		0
X_8	0		0	0	0	0
X_9	1	1	1	1	1	1
X_{12}	0		0	0		0
X_{13}	1	1	1	1		
X_{16}	1	1	1			1

Tabelle 8.28. Kombinationenanalyse der Konjunktion $X_{11} = 1$ und $X_9 = 1$. (s. Tab.8.5).

	Objekte					
#	1	3	6	8	10	16
Klasse	1	1	1	1	2	2
X_1	0	0	0	0	0	0
X_8	0		0	0	0	
X_{12}	0		0	0	0	0
X_{13}	1	1	1	1	1	1
X_{16}	1	1	1			1

Tabelle 8.29. Kombinationenanalyse der Konjunktion $X_{11} = 1$ und $X_{13} = 1$. (s. Tab.8.5).

	Objekte					
#	1	3	4	6	14	16
Klasse	1	1	1	1	2	2
X_1	0	0	0	0	0	0
X_8	0		0	0	0	
X_{12}	0		0	0	0	0
X_{16}	1	1	1	1	1	1

Tabelle 8.30. Kombinationenanalyse der Konjunktion $X_{11} = 1$ und $X_{16} = 1$. (s. Tab.8.5).

	Objekte							
#	1	3	4	6	8	10	14	16
Klasse	1	1	1	1	1	2	2	2
X_8	0		0	0	0	0	0	
X_{12}	0		0	0	0	0	0	0

Tabelle 8.31. Kombinationenanalyse der Konjunktion $X_{11} = 1$ und $X_1 = 0$. (s. Tab.8.5).

	Objekte							
#	1	4	6	8	10	12	14	
Klasse	1	1	1	1	2	2	2	
X_{12}	0	0	0	0	0		0	

Tabelle 8.32. Kombinationenanalyse der Konjunktion $X_{11} = 1$ und $X_8 = 0$. (s. Tab.8.5).

Das nächste zu analysierende Merkmal ist $X_{15} = 1$. Für dieses Merkmal existiert keine einfache Konjunktion, die eine Aufteilung der Klassen ermöglicht. Die Analyse wird mit der Suche nach gültigen Konjunktionen fortgesetzt. Die Merkmale X_1, X_2, X_9, X_{13}, X_{14}, X_{16} können nicht berücksichtigt werden, weil sie die erste Bedingung nicht befriedigen, und die Merkmale X_3, X_7, X_{15} erfüllen nicht die dritte Bedingung. Die übrig gebliebenen Merkmale (die Tabelle) werden sortiert. Die Ergebnisse für das Merkmal $X_{15} = 1$ sind in den Tabellen 8.33 - 8.36 dargestellt.

	Objekte									
#	2	4	5	6	7	8	10	12	14	16
Klasse	1	1	1	1	1	1	2	2	2	2
X_1		0		0		0	0		0	0
X_2		1	1		1		1		1	1
X_3	1	1		1		1	1	1	1	1
X_4	1		1		1	1			1	1
X_5	0	0	0		0	0	0		0	0
X_7	1	1		1		1	1	1	1	1
X_8		0		0	0	0	0	0	0	
X_9				1	1	1		1	1	
X_{12}		0		0	0	0	0		0	0
X_{13}				1	1	1	1			1
X_{14}		1	1		1		1	1	1	
X_{15}	1	1	1	1	1	1	1	1	1	1
X_{16}		1		1	1				1	1

Tabelle 8.33. Kombinationenanalyse der Konjunktion $X_{15} = 1$. (s. Tab.8.5).

Merkmal	X_4	X_5	X_8	X_{12}
Wert	1	0	0	0
In der Klasse 1	4	5	4	4
In der Klasse 2	2	3	3	3

Tabelle 8.34. Die Anzahl der Objekte (Gesichter) entsprechend der Objekteigenschaft und der Klasse für $X_{15} = 1$

	Objekte					
#	2	5	7	8	14	16
Klasse	1	1	1	1	2	2
X_5	0	0	0	0	0	0
X_8			0	0	0	
X_{12}			0	0	0	0

Tabelle 8.35. Kombinationenanalyse der Konjunktion $X_{15} = 1$ und $X_4 = 1$. (s. Tab.8.5).

	Objekte							
#	2	4	5	7	8	10	14	16
Klasse	1	1	1	1	1	2	2	2
X_8		0		0	0	0	0	
X_{12}		0		0	0	0	0	0

Tabelle 8.36. Kombinationenanalyse der Konjunktion $X_{15} = 1$ und $X_5 = 0$. (s. Tab.8.5).

Das nächste zu analysierende Merkmal ist $X_1 = 0$. Für dieses Merkmal existiert keine einfache Konjunktion, die eine Trennung der Klassen ermöglicht. Die Analyse wird mit der Suche nach gültigen Konjunktionen fortgesetzt. Die Merkmale $X_2 – X_5$, X_7, X_{14} können nicht berücksichtigt werden, weil sie die erste Bedingung nicht befriedigen. Die übrigen Merkmale (die Tabelle) werden sortiert. Die Ergebnisse für das Merkmal $X_1 = 0$ sind in den Tabellen 8.37 - 8.42 dargestellt.

	Objekte									
#	1	3	4	6	8	9	10	14	15	16
Klasse	1	1	1	1	1	2	2	2	2	2
X_2	1		1				1	1	1	1
X_3			1	1	1	1	1	1		1
X_4		1			1	1		1	1	1
X_5			0		0	0	0	0	0	0
X_7			1	1	1		1	1	1	1
X_8	0		0	0	0	0	0	0		
X_9	1	1		1	1	1		1		
X_{12}	0		0	0	0		0	0	0	0
X_{13}	1	1		1	1	1	1		1	1
X_{14}	1		1			1	1	1	1	
X_{16}	1	1	1	1		1		1	1	1

Tabelle 8.37. Kombinationenanalyse der Konjunktion $X_1 = 0$ (s. Tab.8.5).

Merkmal	X_9	X_8	X_{12}	X_{13}	X_{16}
Wert	1	0	0	1	1
In der Klasse 1	4	4	4	4	4
In der Klasse 2	2	3	4	4	4

Tabelle 8.38. Die Anzahl der Objekte (Gesichter) entsprechend der Objekteigenschaft und der Klasse für $X_1 = 0$.

	Objekte						
#	1	3	6	8	9	14	
Klasse	1	1	1	1	2	2	
X_8	0		0	0	0	0	
X_{12}	0		0	0		0	
X_{13}	1	1	1	1	1		
X_{16}	1	1	1		1	1	

Tabelle 8.39. Kombinationenanalyse der Konjunktion $X_1 = 0$ und $X_9 = 1$. (s. Tab.8.5).

	Objekte						
#	1	4	6	8	9	10	14
Klasse	1	1	1	1	2	2	2
X_{12}	0	0	0	0		0	0
X_{13}	1		1	1	1	1	
X_{16}	1	1	1		1		1

Tabelle 8.40. Kombinationenanalyse der Konjunktion $X_1 = 0$ und $X_8 = 0$. (s. Tab.8.5).

	Objekte							
#	1	4	6	8	10	14	15	16
Klasse	1	1	1	1	2	2	2	2
X_{13}	1		1	1	1		1	1
X_{16}	1	1	1			1	1	1

Tabelle 8.41. Kombinationenanalyse der Konjunktion $X_1 = 0$ und $X_{12} = 0$. (s. Tab.8.5).

	Objekte							
#	1	3	6	8	9	10	15	16
Klasse	1	1	1	1	2	2	2	2
X_{16}	1	1	1		1		1	1

Tabelle 8.42. Kombinationenanalyse der Konjunktion $X_1 = 0$ und $X_{13} = 1$. (s. Tab.8.5).

Das nächste zu analysierende Merkmal ist $X_4 = 1$. Es gibt keine Merkmale die eine gültige einfache Konjunktion ermöglichen. Deshalb wird die Analyse mit der Suche nach gültigen zweifach Konjunktionen fortgesetzt. Es werden keine Merkmale gefunden, die die erste Bedingung erfüllen. Für $X_5 = 0$ gibt es keine Lösung (Tabelle 8.44).

	Objekte									
#	2	3	5	7	8	9	13	14	15	16
Klasse	1	1	1	1	1	2	2	2	2	2
X_2			1	1			1	1	1	1
X_3	1				1	1		1		1
X_5	0		0	0	0	0		0	0	0
X_7	1				1		1	1	1	1
X_8				0	0	0		0		
X_9		1		1	1	1	1	1		
X_{12}				0	0		0	0	0	0
X_{13}		1		1	1	1	1		1	1
X_{14}			1	1		1		1	1	
X_{16}		1		1		1	1	1	1	1

Tabelle 8.43. Kombinationenanalyse der Konjunktion $X_4 = 1$. (s. Tab.8.5).

	Objekte									
#	2	4	5	7	8	9	10	14	15	16
Klasse	1	1	1	1	1	2	2	2	2	2
X_2		1	1	1			1	1	1	1
X_3	1	1			1	1	1	1		1
X_7	1	1			1		1	1	1	1
X_8		0		0	0	0	0	0		
X_9				1	1	1		1		
X_{12}		0		0	0		0	0	0	0
X_{13}				1	1	1	1		1	1
X_{14}		1	1	1		1	1	1	1	
X_{16}		1		1		1		1	1	1

Tabelle 8.44. Kombinationenanalyse der Konjunktion $X_5 = 0$. (s. Tab.8.5).

Das nächste zu analysierende Merkmal ist $X_8 = 0$. Für dieses Merkmal gibt es keine einfache gültige Konjunktion, die eine Klassifikation ermöglicht. Die Analyse wird mit der Suche von Konjunktionen fortgesetzt. Die Merkmale X_2, X_3, X_7, X_{14} können nicht berücksichtigt werden, weil sie die erste Bedingung nicht befriedigen. Die übrigen Merkmale (siehe Tabelle) werden sortiert. Die Ergebnisse für das Merkmal $X_8 = 0$ sind in den Tabellen 8.45 - 8.49 dargestellt.

	Objekte									
#	1	4	6	7	8	9	10	11	12	14
Klasse	1	1	1	1	1	2	2	2	2	2
X_2	1	1		1			1	1		1
X_3		1	1		1	1	1	1	1	1
X_7		1	1		1		1		1	1
X_9	1		1	1	1	1		1	1	1
X_{12}	0	0	0	0	0		0			0
X_{13}	1		1	1	1	1	1			
X_{14}	1	1		1		1	1	1	1	1
X_{16}	1	1	1	1		1				1

Tabelle 8.45. Kombinationenanalyse der Konjunktion $X_8 = 0$. (s. Tab.8.5).

Merkmal	X_{12}	X_{13}	X_{16}	X_9
Wert	0	1	1	1
In der Klasse 1	5	4	4	4
In der Klasse 2	2	2	2	4

Tabelle 8.46. Die Anzahl der Objekten (Gesichter) entsprechend der Objekteigenschaft und der Klasse für $X_8 = 0$

	Objekte						
#	1	4	6	7	8	10	14
Klasse	1	1	1	1	1	2	2
X_9	1		1	1	1		1
X_{13}	1		1	1	1	1	
X_{16}	1	1	1	1			1

Tabelle 8.47. Kombinationenanalyse der Konjunktion $X_8 = 0$ und $X_{12} = 0$. (s. Tab.8.5).

	Objekte					
#	1	6	7	8	9	10
Klasse	1	1	1	1	2	2
X_9	1	1	1	1	1	
X_{16}	1	1	1		1	

Tabelle 8.48. Kombinationenanalyse der Konjunktion $X_8 = 0$ und $X_{13} = 1$. (s. Tab.8.5).

	Objekte					
#	1	4	6	7	9	14
Klasse	1	1	1	1	2	2
X_9	1		1	1	1	1

Tabelle 8.49. Kombinationenanalyse der Konjunktion $X_8 = 0$ und $X_{16} = 1$. (s. Tab.8.5).

Für $X_8 = 0$ und $X_{12} = 0$» kann „$X_{13} = 1$ und $X_9 = 1$" die Klassen trennen. Die Konjunktion

«$X_8 = 0$ und $X_{12} = 0$ und $X_{13} = 1$ und $X_9 = 1$" (7)

wird protokolliert. Für die übrigen Merkmale gibt es keine Lösung.

Das nächste zu analysierende Merkmal ist $X_9 = 1$. Für dieses Merkmal existiert keine einfache Konjunktion, die eine Klassentrennung ermöglicht. Die Analyse wird mit Suche nach gültigen Konjunktionen fortgesetzt. Die Merkmale X_2, X_3, X_7, X_{14} können nicht berücksichtigt werden, weil sie die erste Bedingung nicht befriedigen. Die übrigen Merkmale (siehe Tabelle) werden sortiert. Die Ergebnisse für das Merkmal $X_9 = 1$ sind in den Tabellen 8.50- 8.53 dargestellt. Es gibt keine Lösung.

	Objekte									
#	1	3	6	7	8	9	11	12	13	14
Klasse	1	1	1	1	1	2	2	2	2	2
X_2	1			1			1		1	1
X_3			1		1	1	1	1		1
X_7			1		1			1	1	1
X_{12}	0		0	0	0				0	0
X_{13}	1	1	1	1	1	1			1	
X_{14}	1			1		1	1	1		1
X_{16}	1	1	1	1		1			1	1

Tabelle 8.50. Kombinationenanalyse der Konjunktion $X_9 = 1$. (s. Tab.8.5).

Merkmal	X_{13}	X_{12}	X_{16}
Wert	1	0	1
In der Klasse 1	5	4	4
In der Klasse 2	2	2	3

Tabelle 8.51. Die Anzahl der Objekte (Gesichter) entsprechend der Objekteigenschaft und der Klasse für $X_8 = 0$

	Objekte						
#	1	3	6	7	8	9	13
Klasse	1	1	1	1	1	2	2
X_{12}	0		0	0	0		0
X_{16}	1	1	1	1		1	1

Tabelle 8.52. Kombinationenanalyse der Konjunktion $X_9 = 1$ und $X_{13} = 1$. (s. Tab.8.5).

	Objekte					
#	1	6	7	8	13	14
Klasse	1	1	1	1	2	2
X_{16}	1	1	1		1	1

Tabelle 8.53. Kombinationenanalyse der Konjunktion $X_9 = 1$ und $X_{12} = 0$. (s. Tab.8.5).

Das nächste zu analysierende Merkmal ist $X_{12} = 0$. Für dieses Merkmal existiert keine gültige einfach Konjunktion, die eine Trennung der Klassen ermöglicht. Die Analyse wird mit der Suche nach gültigen Konjunktionen fortgesetzt. Für X_{13}, gibt es keine Lösung. Die anderen Merkmale können nicht berücksichtigt werden, weil sie die Bedingung nicht befriedigen. Die Ergebnisse für das Merkmal $X_{12} = 0$ sind in den Tabellen 8.54 und 8.55 dargestellt. Für die restlichen Merkmale $X_{13} = 1$, $X_{16} = 1$, $X_2 = 1$, $X_3 = 1$, und $X_7 = 1$ gibt es keine Lösung (s. Tabellen 8.56 - 8.60).

	Objekte									
#	1	4	6	7	8	10	13	14	15	16
Klasse	1	1	1	1	1	2	2	2	2	2
X_2	1	1		1		1	1	1	1	1
X_3		1	1		1	1		1		1
X_7		1	1		1	1	1	1	1	1
X_{13}	1		1	1	1	1	1		1	1
X_{14}	1	1		1		1		1	1	
X_{16}	1	1	1	1			1	1	1	1

Tabelle 8.54. Kombinationenanalyse der Konjunktion $X_{12} = 0$. (s. Tab.8.5).

	Objekte							
#	1	6	7	8	10	13	15	16
Klasse	1	1	1	1	2	2	2	2
X_{16}	1	1	1			1	1	1

Tabelle 8.55. Kombinationenanalyse der Konjunktion $X_{12} = 0$ und $X_{13} = 1$. (s. Tab.8.5).

	Objekte									
#	1	3	6	7	8	9	10	13	15	16
Klasse	1	1	1	1	1	2	2	2	2	2
X_2	1			1			1	1	1	1
X_3			1		1	1	1			1
X_7			1		1		1	1	1	1
X_{14}	1			1		1	1		1	
X_{16}	1	1	1	1		1		1	1	1

Tabelle 8.56. Kombinationenanalyse der Konjunktion $X_{13} = 1$. (s. Tab.8.5).

	Objekte									
#	1	3	4	6	7	9	13	14	15	16
Klasse	1	1	1	1	1	2	2	2	2	2
X_2	1		1		1		1	1	1	1
X_3			1	1		1		1		1
X_7			1	1			1	1	1	1
X_{14}	1		1		1	1		1	1	

Tabelle 8.57. Kombinationenanalyse der Konjunktion $X_{16} = 1$. (s. Tab.8.5).

	Objekte									
#	1	4	5	7	10	11	13	14	15	16
Klasse	1	1	1	1	2	2	2	2	2	2
X_3		1			1	1		1		1
X_7		1			1		1	1	1	1
X_{14}	1	1	1	1	1	1		1	1	

Tabelle 8.58. Kombinationenanalyse der Konjunktion $X_2 = 1$. (s. Tab.8.5).

	Objekte									
#	2	4	6	8	9	10	11	12	14	16
Klasse	1	1	1	1	2	2	2	2	2	2
X_7	1	1	1	1		1		1	1	1
X_{14}		1			1	1	1	1	1	

Tabelle 8.59. Kombinationenanalyse der Konjunktion $X_3 = 1$. (s. Tab.8.5).

	Objekte									
#	2	4	6	8	10	12	13	14	15	16
Klasse	1	1	1	1	2	2	2	2	2	2
X_{14}		1			1	1		1	1	

Tabelle 8.60. Kombinationenanalyse der Konjunktion $X_7 = 1$. (s. Tab.8.5).

Für den betrachteten Datensatz konnten für die erste Klasse 7 logische Regeln gefunden werden (s. Tab. 8.61). Für die zweite Klasse findet man durch analoges Vorgehen 5 Regeln. Die Gesamtanzahl der Regeln beträgt 12. In Abb. 8.2 ist die Protokolldatei des Programms *DataControl* (ein Teilprogramm von *GeneRule* 2.0) dargestellt. Im nächsten Verarbeitungsschritt wird die extrahierte Regelbasis vereinfacht.

Erste Klasse	Zweite Klasse
Regel 1: X_3=0 und X_7 = 0	**Regel 1:** X_2 = 1 und X_7= 1 und X_4 = 1
Regel 2: X_2=0 und X_{14} = 0	**Regel 2:** X_2 = 1 und X_7= 1 und X_{13} = 1
Regel 3: X_6 = 1 und X_{11}= 1 und X_9 = 1	**Regel 3:** X_3 = 1 und X_{14}= 1 und X_9 = 1
Regel 4: X_6 = 1 und X_{11}= 1 und X_{16} = 1	**Regel 4:** X_7 = 1 und X_4= 1 und X_{16} = 1
Regel 5: X_{10} = 1 und X_{15}= 1 und X_4 =1	**Regel 5:** X_1 = 0 und X_4= 1 und X_5 =0 und X_{16} = 1
Regel 6: X_{11} = 1 und X_9= 1 und X_{13} =1	
Regel 7: X_8 = 0 und X_{12}= 0 und X_{13} =1 und X_9 =1	

Tabelle 8.61. Extrahierte logische Regeln

In der ersten Klasse können die Objekte 1, 3, 4 und 6 durch Regel 4 erkannt werden. Die übrigen Objekte 2, 5, 7 und 8 werden durch die Regel 5 erkannt. In der zweiten Klasse können alle Objekte durch die Regel 3, die für die Objekte 9, 11, 12 und 14 gültig ist, und die Regel 2, die für die Objekte 10, 13, 15 und 16 gültig ist, erkannt werden.

```
#Knowledge base name: djuck
#Data base name: text_djuck_trans.dat
Vector of 2: 9 Recognized by rules for 2:2 1: 0 -> class 2          By rules 2: 3 5
Vector of 2: 10 Recognized by rules for 2:1 1: 0 -> class 2         By rules 2: 2
Vector of 2: 11 Recognized by rules for 2:1 1: 0 -> class 2         By rules 2: 3
Vector of 2: 12 Recognized by rules for 2:1 1: 0 -> class 2         By rules 2: 3
Vector of 2: 13 Recognized by rules for 2:3 1: 0 -> class 2         By rules 2: 1 2 4
Vector of 2: 14 Recognized by rules for 2:4 1: 0 -> class 2         By rules 2: 1 3 4 5
Vector of 2: 15 Recognized by rules for 2:4 1: 0 -> class 2         By rules 2: 1 2 4 5
Vector of 2: 16 Recognized by rules for 2:4 1: 0 -> class 2         By rules 2: 1 2 4 5
 Recognized true 8, error 0, unk 0 of 8 vectors of 2
Vector of 1: 1 Recognized by rules for 1:5 2: 0 -> class 1          By rules 1: 1 3 4 6 7
Vector of 1: 2 Recognized by rules for 1:2 2: 0 -> class 1          By rules 1: 2 5
Vector of 1: 3 Recognized by rules for 1:5 2: 0 -> class 1          By rules 1: 1 2 3 4 6
Vector of 1: 4 Recognized by rules for 1:1 2: 0 -> class 1          By rules 1: 4
Vector of 1: 5 Recognized by rules for 1:2 2: 0 -> class 1          By rules 1: 1 5
Vector of 1: 6 Recognized by rules for 1:5 2: 0 -> class 1          By rules 1: 2 3 4 6 7
Vector of 1: 7 Recognized by rules for 1:3 2: 0 -> class 1          By rules 1: 1 5 7
Vector of 1: 8 Recognized by rules for 1:5 2: 0 -> class 1          By rules 1: 2 3 5 6 7
 Recognized true 8, error 0, unk 0 of 8 vectors of 1
```

Abbildung 8.2. Log-Datei des Programms *DataControl* (*GeneRule* 2.0)

Auf diese Weise kann die Regelanzahl von 12 auf 4 reduziert werden.

ANHANG B

LEBENSLAUF

23.07.1966	geboren in Leningrad (St.Petersburg) in Russland
09-73 bis 06-83	allgemein bildende Schule, Leningrad
06-83	Hochschulreife
09-83 bis 06-89	Pädiatrische Akademie, Leningrad. Abschluss: Diplom - Arzt
02-90 bis 06-94	Wissenschaftlicher Mitarbeiter im Wissenschaftlichen Zentrum des Ministeriums für Gesundheit der Russischen Föderation in der Pädiatrische Akademie, St. Petersburg. Fachbereich: (bio)medizinische Informatik
09-87 bis 06-91	Nord-West Polytechnische Universität (Extern) Leningrad Abschluss: Diplom - Ingenieur für Elektronische Rechentechnik
10-91 bis 04-94	Promotionsstudium an der Nord-West Polytechnische Universität, St. Petersburg
04-94	Promotion an der Staatlichen Elektrotechnischen Universität, St. Petersburg Thema: „Symptomokomplexes Herangehen bei der Ausarbeitung eines Expertensystems bei der Prognostizierung der Entwicklung des Krankheitsverlaufs in der Neonatologie“ Abschluss: Kandidat der technischen Wissenschaften (Doktorgrad)
07-94 bis 05-96	Arzt und Ingenieur für Labortechnik im Zentrallabor des Staatlichen Krankenhauses Nr. 2, St. Petersburg
09-95 bis 11-95	Weiterbildung. Klinische Biochemie. MAPO, St. Petersburg.
06-96	Einreise in die Bundesrepublik Deutschland
09-96 bis 02-97	Teilnahme an einem Sprachkurs für Deutsch, Bad Langensalza (Thueringen).
04-98 bis 03-99	Stipendiat der Otto-Benecke-Stiftung, Praktikumsstelle in der Fachhochschule Jena
10-99 bis 06-00	Teilnahme an einem Kurs „Netzwerkspezialist“, IAD, Erfurt
07-00 bis 12-01	Softwareentwickler, MPSeCOM, Jena
01-02 bis 12-04	Hans-Knöll-Institut Jena Promotionsarbeit

Jena,

Publikationen

Artikel

1. V.I. Gutkin, V.L. Monossov, A.G. Francuz: "*Intelligence system in reanimation and an intensive care*". North-West polytechnic university – St.-Petersburg, 1992. VINITI 01.04.92., 838-B92.
2. V.I. Gutkin, V.L. Monossov, A.G. Francuz: "*Modern methods of development of knowledge bases of expert systems*". North-West polytechnic university - St.-Petersburg, 1993. VINITI 07.04.93, 860-B93.
3. V.I. Gutkin, V.L. Monossov, A.G. Francuz: "*Modern systems of an artificial intelligence*". North-West polytechnic university - St.-Petersburg, 1993- VINITI 07.04.93, 861-B93.
4. V.I. Gutkin, V.L. Monossov, A.G. Francuz: "*Problems of expert systems*". North-West polytechnic university - St.-Petersburg, 1993. – VINITI 07.04.93, 862-B93.
5. V.I. Gutkin, V.L. Monossov, A.G. Francuz: "*The shell of the prognostic expert systems: the PROLOG application for medical diagnostic*". North-West polytechnic university - St.-Petersburg, 1993. – VINITI 07.04.93, M 863-B93.
6. V.L. Monossov: The PhD. thesis. St. Petersburg state electro technical university. St.-Petersburg. 1994.
7. V.L. Monossov: "*Symptomocomplex approach at development of the expert systems for prediction of severity of illness in a neonatology*". St. Petersburg state electro technical university. Dissertation. St.-Petersburg. 1994.

Eingereichte Publikation:

Monossov, V., Weller, K., Guthke, R.: Rule-Based Knowledge Discovery by Gene Expression Analysis. Computers in Biology and Medicine, submitted 2004

Vorträge

1. V.L. Monossov, "*Characteristic of modern systems of severity of illness development prediction*". «*Technical diagnosing* – 93», - St.-Petersburg. 8-10 June 1993. Seiten 135−136.
2. V.L. Monossov, "*The shell of the prognostic expert system*". «**Technical diagnosing**–93», - St.-Petersburg. 8-10 June 1993, - Seiten 133- 134.

Poster

1. V. Monossov., R. Guthke. GeneRule: "*A Tool for Extraction and Validation of Knowledge for Rule-Based Expert Systems*". Saarbrueken. 2002.
2. V. Monossov, R. Guthke, M. Pfaff: "*Extraction and Validation of Rule-based Knowledge from Gene Expression Data*". Muenich. 2003.

Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit selbständig, ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel und Literatur angefertigt habe.
Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.
Weitere Personen waren an der inhaltlich-materiellen Erstellung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe von Vermittlungs- bzw. Beratungsdiensten (Promotionsberater oder andre Personen) in Anspruch genommen. Niemand hat von mir unmittelbar oder mittelbar geldwerte Leistungen für Arbeit erhalten, die im Zusammenhang mit dem Inhalt der vorliegenden Dissertation stehen.
Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.
Die geltende Promotionsordnung der Biologisch-pharmazeutischen Fakultät ist mir bekannt.
Ich versichere ehrenwörtlich, dass ich nach bestem Wissen die reine Wahrheit gesagt und nichts verschwiegen habe.

Jena, den
(Vladimir Monossov)

Danksagung

Diese Arbeit wurde in der Zeit von Januar 1992 bis Dezember 2004 am Hans-Knöll-Institut für Naturstoff-Forschung e. V. (HKI) in der Abteilung Angewandte Mikrobiologie (AMB) angefertigt.
Zuallererst mochte ich mich herzlich bei Herrn Dr. habil. Reinhard Guthke für die Möglichkeit bedanken, in seiner Abteilung diese Dissertation anzufertigen. Seine Unterstützung und seine stete Diskussionsbereitschaft waren mir sehr wertvoll. Er hat, nicht zuletzt durch die kritische Revision des Manuskripts, maßgeblich zum Gelingen der Arbeit beigetragen.
Herrn Prof. Dr. Stefan Schuster (Friedrich-Schiller-Universität Jena) habe ich für seine Unterstützung und Herrn Prof. Dr. Stefan Wölfl (Ruprecht-Karls-Universität Heidelberg) und Mitgliedern seiner Arbeitsgruppe an der Universität Jena für viele Gespräche und Hinweise zu danken.
Mein ganz besonderer Dank gilt Dipl.-Ing. Wolfgang Schmidt-Heck (HKI/AMB) für seine Unterstutzung.
Für weitere interessante Anregungen und Diskussionen möchte Dr. Martin Hoffmann (HKI/AMB) danken.